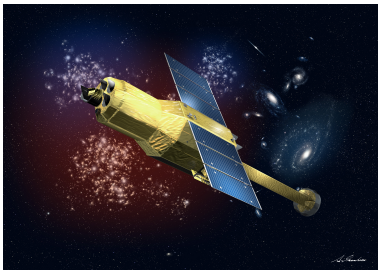

HITOMI

TASKS HELP

As included with the Hitomi Software version 6

Date: May 2017

X-ray Astrophysics Laboratory (NASA/Goddard Space Flight Center)
And the
Institute of Space and Astronautical Science (ISAS/JAXA)

NAME aberattitude - Correct an attitude file for aberration effects

USAGE aberattitude infile outfile

DESCRIPTION
'aberattitude' is a mission-independent tool, which modifies an attitude file to correct for aberration effects. The task requires an attitude input file (parameter 'infile') and computes the correction for each unique time present in this input file. The output file, specified by the parameter 'outfile' contains the corrected attitude columns along the original ones renamed with the string '_OLD' appended. For example, for an input file with a 'QPARAM' attitude column, the output file has a column named 'QPARAM_OLD' containing the original values and a column named 'QPARAM' containing the corrected values. The aberration of light (also referred to as stellar aberration) is a phenomenon producing an apparent motion of celestial objects about their locations dependent on the velocity of the observer. If the parameter 'debug' is set to 'yes', the following quantities are printed for each row of the input table: 'TIME', mission time; 'VTOT', total spacecraft velocity; 'VSATX', 'VSATY', and 'VSATZ', X, Y and Z components of the total satellite velocity; 'VEARTH', Earth velocity; 'VEARTHX', 'VEARTHY', and 'VEARTHZ', X, Y and Z components of the Earth velocity; 'ERADVEL', projection of Earth velocity onto unit vector toward the observed target. The (X, Y, Z) system is a right-handed Equatorial system where X points toward the Vernal Equinox and Z toward the North Celestial Pole. Velocity unit is km/s.

PARAMETERS
infile [filename]
Name of the input attitude file.

outfile [filename]
Name of the output attitude file.

(orbfile = NONE) [filename]
Name of the orbit file used to retrieve spacecraft velocity in Earth-Centered Inertial (ECI) Cartesian coordinates. This file is required only if 'orbaber=yes'.

(alignfile = NONE) [filename]
Name of the spacecraft alignment calibration file. The alignment file specifies the rotation between the telescope and spacecraft axes.

(annaber = yes) [string]
'annaber' allows to correct for annual aberration. The allowed setting are yes, no or invert. If set to no, the default, no correction is applied. If set to yes, the effects of annual aberration are taken into account when calculating the SKY coordinate values. If set to invert, multiplies the annual aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Annual aberration is the apparent bending of light due to the Earth's orbit around the Sun. This is at most a ~20.49 arcsec effect.

(orbaber = no) [string]
'orbaber' allows to correct for orbital aberration. The allowed setting are yes, no or invert. If set to no, the default, the orbital aberration is not corrected. If set to yes, the effects of orbital aberration are taken into account when calculating the SKY coordinates, provided an input orbit file is specified using the parameter 'orbfile'. If set to 'invert', multiplies the orbital aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Orbital aberration is the apparent bending of light due to the satellite's orbit around the Earth. For a satellite in low-earth orbit this is at most a ~5 arcsec effect.

(attext = ATTITUDE) [string]
Name of the FITS binary table extension of the input attitude file containing the attitude to be corrected. The attitude in the input file can be in one of three following formats specified by the parameter 'attform' (see below).
1. attform=QUAT: Quaternion [x, y, z, real]
2. attform=EULER: Z-Y-Z Euler angles (phi, theta, psi) in degrees.
3. attform=POINTING: Pointing angles (right ascension, declination, roll) in degrees.

If the attitude file contains separate columns giving the attitude in more than one format, only the column formatted as specified with the 'attform' parameter is corrected.

(attcol = QPARAM) [string]
Name of the column in the input attitude file containing the attitude to be corrected. The data in this column must be in the format given by the 'attform' parameter.

(attform = QUAT) [string]
Format of the attitude table. Three formats are supported. The QUAT format is a quaternion [x, y, z, real]. The EULER format is a Z-Y-Z Euler angle trio [phi, theta, psi] in degrees and the POINTING format.

(orbext = ORBIT) [string]
Name of the FITS binary table extension of the orbit file containing the velocity vectors.

(orbcol = VELOCITY) [string]
Name(s) of the FITS column(s) containing orbit values in the orbext extension of the orbit file. This parameter is linked to the parameter 'orbform'. The only acceptable values for 'orbcol' are: VELOCITY, VECTOR or COMPONENTS. The default is 'VELOCITY'. If 'orbform=VECTOR', then 'orbcol' is the name of the single FITS column containing the orbit values as a vector. If 'orbform=COMPONENTS', then 'orbcol' must be a string containing three comma-separated column names, specifying in order the X, Y and Z components of the orbital velocity. These columns must be scalar. Velocity must be in km/s in an Earth-Centered Inertial system.

(orbform = VECTOR) [string]
Format of the orbital velocity column or columns in the orbit file. Three formats are supported. For the VECTOR format (DEFAULT), the velocity is provided as a vector column with three elements (X, Y and Z in Earth-Centered Inertial (ECI) system). For the COMPONENT format, the velocity is provided in three separate columns.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

(clobber = no) [boolean]
Overwrites the existing output file if set to yes (yes/[no]).

(chatter = 1) [integer]
Chatter level for output. Set to 0 to suppress output, or to 1, 2, or 3 for increasing the chatter of the output.

(logfile = !DEFAULT) [string]
Log filename. If set to DEFAULT uses the name of the task and, if preceded by '!', overwrite the file if it exists. If set to NONE no log file is created.

(debug = no) [boolean]
Diagnostic output is printed out on the screen if set to yes (yes/[no]).

(history = yes) [boolean]
Records tool parameters in HISTORY ([yes]/no).

(mode = ql) [string ql|hl|q]
Mode to query the parameter file. Acceptable values include: "ql (query and learn/remember), "hl" (hidden and learn/remember), "q" (query but don't remember), "h" (hidden).[1]

EXAMPLES

1. Modify an attitude file containing the attitude in the quaternion format, correcting only for the annual aberration.
aberattitude infile="attitude.att" outfile="attitude-corr.att" alignfile="align.fits" annaber=yes clobber=yes
2. Modify an attitude file containing the attitude in the Euler angle format, correcting for the annual and orbital aberration.
aberattitude infile="attitude.att" outfile="attitude-corr.att" alignfile="align.fits" annaber=yes orbaber=yes attform=EULER \
attcol="EULER" orbfile="orbit.fits" orbext="SAT_ORBIT" orbcol="SAT_VELOCITY" orbform=VECTOR clobber=yes
---
NAME aberposition - Correct coordinates for aberration effects.

USAGE aberposition ra dec time

DESCRIPTION
'aberposition' is a mission-independent tool that corrects the coordinates for aberration effects. The task requires an initial RA and Dec (given in degrees) and a time (UTC format). To correct for orbital aberration, an orbit file is necessary. If no orbit file is specified, the correction includes only the annual correction. The task outputs the corrected RA and Dec in the 'outra' and 'outdec' parameters.

PARAMETERS
ra [real]
Input Right Ascension in degrees. This must be a value between 0 and 360.

---

[1] Because these are always present in tasks they are omitted in the reminder of the document and replaced by the acronym of the flags "**ccldhm**"

dec [real]
Input Declination in degrees. This must be a value between -90 and +90.

time [string]
Input UTC date and time in the format "yyyy-mm-ddThh:mm:ss", e.g. "2014-08-01T12:00:00".

(orbfile = NONE) [string]
Name of the orbit file used to retrieve spacecraft velocity in Earth-Centered Inertial (ECI) Cartesian coordinates. Required for orbital aberration corrections

(annaber = no) [string]
'annaber' enables annual aberration correction. The allowed setting are yes, no or invert. If set to no, the default, no correction is applied. If set to yes, the effects of annual aberration are taken into account when calculating the SKY coordinate values. If set to invert, multiplies the annual aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Annual aberration is the apparent bending of light due to the Earth's orbit around the Sun. This is at most a ~20.49 arcsec effect.

(orbaber = no) [string]
'orbaber' enables orbital aberration correction. The allowed setting are yes, no or invert. If set to no, the default, the orbital aberration is not corrected. If set to yes, the effects of orbital aberration are taken into account when calculating the SKY coordinates and an orbit file is required (parameter 'orbfile'). parameter 'orbfile'. If set to 'invert', multiplies the orbital aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Orbital aberration is the apparent bending of light due to the satellite's orbit around the Earth. For a satellite in low-earth orbit this is at most a ~5 arcsec effect.

(leapsecfile = REFDATA) [string]
Name of the FITS file containing the table of leap seconds. This file is needed in order to accurately convert UTC to the time system used in the orbit file. The values for this flag can be CALDB, REFDATA or a filename If the parameter is set to CALDB, the default, the file is read from the calibration database; if REFDATA, the file is accessed in the reference data area of the HEASoft installation.

(orbext = ORBIT) [string]
Name of the FITS binary table extension of the orbit file containing the velocity vectors.

(orbcol = VELOCITY) [string]
Name(s) of the FITS column(s) containing orbit values in the orbext extension of the orbit file. This parameter is linked to the parameter 'orbform'. The only acceptable values for 'orbcol' are: VELOCITY, VECTOR or COMPONENTS. The default is 'VELOCITY'. If 'orbform=VECTOR', then 'orbcol' is the name of the single FITS column containing the orbit values as a vector. If 'orbform=COMPONENTS', then 'orbcol' must be a string containing three comma-separated column names, specifying in order the X, Y and Z components of the orbital velocity. These columns must be scalar. Velocity must be in km/s in an Earth-Centered Inertial system.

(orbform = VECTOR) [string]
Format of the orbital velocity column or columns in the orbit file. Three formats are supported. For the VECTOR format (DEFAULT), the velocity is provided as a vector column with three elements (X, Y and Z in Earth-Centered Inertial (ECI) system). For the COMPONENT format, the velocity is provided in three separate columns.

(outra = 0.0) [real]
The adjusted value of Right Ascension is stored in this parameter in the .par file on exit from the tool.

(outdec = 90.0) [real]
The adjusted value of Declination is stored in this parameter in the .par file on exit from the tool.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Derive the annual aberration for a given pointing and time using the leapsecond file extracted from CALDB. Chatter set to 2 prints out the output.
aberposition ra=161.2649625 dec=-59.68451667 time=2009-06-10T01:49:30 chatter=2

2. Derive the annual and orbital aberration for a given pointing and time. The orbit file has the orbit velocity stored in three separate columns.
aberposition ra=161.2649625 dec=-59.68451667 time=2009-06-10T01:49:30 leapsecfile="CALDB" orbaber=yes orbfile="orbit.fits" \
orbcol="VX,VY,VZ" orbform=COMPONENTS chatter=2

NAME aharfgen - Make an ancillary response function (ARF) file for the SXS or SXI, or a response matrix (RSP) file for the HXI

USAGE aharfgen xrtevtfile source_ra source_dec instrume dattfile emapfile qefile contamifile gateevalvefile sampling rmffile erange onaxisffile onaxiscfile outfile regmode regionfile mirrorfile obstructfile frontreffile backreffile pcolreffile scatterfile numphoton sourcetype imgfile

DESCRIPTION
'aharfgen' is a script that runs the tools 'ahsxtarfgen' and 'hxirspeffimg'. When 'aharfgen' is used to run 'ahsxtarfgen', an ancillary response function (ARF) file for the SXS + SXT-S (parameter 'instrume=SXS') or the SXI + SXT-I ('instrume=SXI') combination is created. When 'aharfgen' is used to run 'hxirspeffimg', a spectral response matrix (.RSP file) for the HXI1 + HXT ('instrume=HXI1') or HXI2 + HXT ('instrume=HXI2') combination is created. 'aharfgen' takes as input an exposure map file created by the tool 'ahexpmap'. The exposure map file contains information about the satellite attitude and variation in the telescope optical axis pointing, as well as spatial effective exposure time information. A standard SAO region file input can be used to specify the region in RA/DEC or DET coordinates for which the output ARF or RSP files are created (see 'regmode' and 'regionfile' parameter). If the region is in RA/DEC coordinates, the script then creates a number of selection regions in the DET coordinate system corresponding to discrete satellite attitude intervals (or histogram bins) described in the exposure map file. The options for specifying the spatial distribution of the of X-ray source, are (1) point source ('sourcetype=point'), (2) uniform source ('sourcetype=flatcircle') with a set radius (see 'flatradius' parameter), (3) beta model ('sourcetype=betamodel'; see 'betapars' parameter), and (4) input image ('sourcetype='image'). The image option, useful for extended sources, uses an input image file to create a photon event list by the simulation tool 'heasim'. This event list is subsequently input into the raytracing code. The input image file could be made from data from a different mission, or from a model. The source position is input using the 'source_ra' and 'source_dec' parameters.
'aharfgen' runs the raytracing tool 'xrtraytrace', using information about the source and region selection to simulate photon paths through the X-ray telescope. The photon paths, or events, along with the source and region information, are passed onto 'ahsxtarfgen' or 'hxieffrspimg' in order to enable the telescope effective area to be calculated. Detector efficiencies are calculated in 'ahsxtarfgen' and 'hxieffrspimg', not in 'aharfgen'. The raytracing file is saved and can be used again to run either 'ahsxtarfgen' or 'hxieffrspimg' without running 'xrtraytrace'. When 'aharfgen' is run, and it finds that a raytracing file with the requested file name already exists, 'aharfgen' skips running the raytracing code again, and instead pass on the existing event file onto 'ahsxtarfgen' or 'hxirspeffimg'. Running the raytracing code is very time-intensive so this feature avoids unnecessarily long run times whenever possible. The event file from previous runs of the raytracing may be used if the arf is recalculated using a different binning (the binning is defined by rmf), different region and whether or not the auxiliary transmission file is used.
The number of photons injected into the raytracing code determines the statistical accuracy of the effective area and detector efficiencies. This number is controlled by the parameter 'numphoton'. The total number of photons injected into the raytracing code is determined by an algorithm that takes into account the relative time intervals for each attitude bin as given in the first extension of the exposure map file. Each row in the first extension of the exposure map file corresponds to one attitude bin. The value of the parameter 'numphoton' corresponds roughly to the number of raytracing photons allocated to each attitude bin per energy grid point. If any attitude bin has a time interval significantly larger than the average time interval, 'aharfgen' allocates additional raytracing photons to that attitude position. Note that the raytracing is performed on a coarse energy grid for modeling the spatial distribution of photons, which has a weak energy dependence. The final effective area calculated by 'aharfgen' combines these coarse-grid results with some pre-calculated raytracing results that were obtained on a fine energy grid that captures all of the atomic physics at a level of detail sufficient for the SXS.
As a rough guide, one should aim for a minimum total of 3 million photons over the whole energy range, and numphoton can be estimated by numphoton ~ (total number of photons/number of attitude bins/X), where X~16 for SXS or SXI over the energy range 0.4-15 keV, and X~270 for the HXI over the energy range 4-70 keV. The run time for the raytracing code is approximately 1 minute per 100,000 photons. The energy range for which 'aharfgen' calculates the ARF or RSP files is selectable by the user, and ranges wider than the nominal ranges just mentioned may be specified. For the SXS the energy range can span 0.03 to 30 keV; and, for the SXI, 0.03 to 24 keV. For the HXI the energy range can span 2 to 120 keV. However the calibration outside of the nominal energy ranges is not reliable, so it is not recommended to choose wider ranges the nominal ones, which would also result in a substantial increase in the runtime.

PARAMETERS
xrtevtfile [filename]
Name of event/history file created by the raytracing program 'xrtraytrace', and used by 'aharfgen'. If xrtevtfile does not exist 'aharfgen' creates it by running 'xrtraytrace'. If xrtevtfile is the name of an existing file that was previously created by 'aharfgen', this file is used instead of creating a new one.

source_ra [real]

R. A. (degrees) of the X-ray source for which the ARF is to be created. For 'sourcetype=flatcircle' or 'betamodel', it is the RA of the center of the source. For 'sourcetype=image', the parameter 'source_ra' represents the RA position of the source within the image. If there is no obvious source, it should be set to the RA position of the center of the image.

source_dec [real]
Declination (DEC, degrees) of the X-ray source for which the ARF is to be created. For 'sourcetype=flatcircle' or 'betamodel', it is the DEC value of the center of the source. For 'sourcetype=image', the parameter 'source_dec' represents the DEC position of the source within the image. If there is no obvious source in the image, it should be set to the DEC position of the center of the image.

(nompntpars) [string]
This parameter is only used if no exposure map is input to 'aharfgen' ('emapfile=NONE'). The parameter consists of a string of five numbers as follows: (1) RA of satellite pointing direction; (2) DEC of satellite pointing direction; (3) Roll angle of satellite; (4) RA of the telescope optical axis; (5) DEC of the telescope optical axis.

telescop [string]
Mission name (value to write in header keyword TELESCOP for CALDB).

instrume [string]
Instrument for which the ARF is to be made: SXI, SXS, HXI1, or HXI2.

(teldeffile) [filename]
Name of the telescope definition file appropriate for the instrument for which the ARF is to be made. If the parameter is set to CALDB, the file is read from the calibration database.

dattfile [filename]
Name of the CAMS delta-attitude file describing the position of the optical axis in RAWX, RAWY, as a function of time. This parameter is only used if 'instrume = HXI1' or 'instrume = HXI2'. The columns used are RAWX_FLOAT, RAWY_FLOAT.

filtoffsetfile [filename]
Name of the delta-attitude file describing the CAMS motion with columns DELTARAWX, DELTARAWY, SINANGLE, COSANGLE, as a function of time. The file is made by running cams2att with 'filtoffset=filtoffsetfile'.

emapfile [filename]
Name of the exposure map file (made by the tool 'ahexpmap') containing histograms of satellite attitude and related quantities, as well as good time intervals (GTI) for the attitude bins. In the case of the SXI or SXS there is also an effective exposure time image in the primary extension of the exposure map.

(qefile) [filename]
Name of the file containing the quantum efficiency (QE) for the detector. If the parameter is set to CALDB, the file is read from the calibration database. For the SXI the QE is combined with the optical blocking layer (OBL) transmission.

(obffile) [filename]
Name of the optical blocking filter file (needed for SXS only). If the parameter is set to CALDB, the file is read from the calibration database.

(fwfile) [filename]
Name of the filter wheel filter file (needed for SXS only). If the parameter is set to CALDB, the file is read from the calibration database.

contamifile [filename]
Name of the file containing information to calculate the transmission due to contaminants on the detector as a function of time, energy, and detector position. If the parameter is set to CALDB, the file is read from the calibration database.

(abund = "1.0") [string]
Relative abundances of contaminants. This number multiplies the abundances of all of the contaminant materials in the calibration file.

(cols = "0.0") [string]
Additional column densities for contaminants (1E18 cm^-2). The column densities of all of the contaminant materials in the calibration file are modified by adding the value of this parameter to the value in contamifile.

(covfac = "1.0") [string]
Partial covering factors for contaminant materials. This number multiplies the partial covering factors of all of the contaminant materials in the calibration file.

gatevalvefile [filename]
Name of the SXS gate valve calibration file. This file is only needed for SXS observations for which the value of the GATEVALV keyword is equal to CLOSED. The file accounts for the blocking and attenuation effects of the gatevalve.

sampling [integer]
This parameter is only needed if 'instrume = HXI1' or 'instrume = HXI2'. It is the sampling factor for the HXI CAMS delta-attitude bins. if 'sampling = 1' then the original binning in the 'dattfile' is used. If 'sampling = N', then use the first bin, then the (N+1)th bin etc.

(baffle = "64.0 64.0 25.35 622.0 24.67 124.0") [string]
This parameter is only needed if 'instrume =HXI1' or 'instrume = HXI2'. It is a string list of six numbers that specify parameters of the HXI baffle model as follows: (1) Baffle center RAWX coordinate, (2) Baffle center RAWY coordinate, (3) Radius of top entrance of baffle (mm), (4) Height of baffle entrance hole above focal plane (mm), (5) Radius of bottom exit of baffle (mm), (6) Height of exit hole above focal plane (mm).

rmffile [filename]
Name of a response matrix file (RMF) valid for the instrument for which the arf is generated. For the SXI and SXS the rmffile is always a single file, generated either by 'sxsrmf' or 'sxsirmf,' from which the energy grid is read and used within aharfgen for the output energy grif of the ARF file. For the HXI instead the rmffile may be input as ascii file containing a list of the rmffiles, prefixed with "@" or as CALDB. The ascii file contains one RMF filename per line and to calculate the ARF the HXI requires 5 RMF one for each of the detector layers. If set to CALDB aharfgen automatically retrives the correct RMF.

erange [string]
A string containing four numbers. The first and second numbers correspond to the minimum and maximum energy, respectively, of the valid data in the output ARF or RSP file (points outside this range are filled with zeros). Restricting the energy range in this way may be used to shorten the run time, if a restricted energy range is sufficient for the application. The third and fourth numbers in the input parameter string are only relevant if 'sourcetype = image' (one of the extended source options), and are set to the lower and upper energy bound respectively, of the image data.

onaxisffile [filename]
Name of the file and extension that contains the on-axis telescope effective area, appropriate for the instrument, pre-calculated on a fine energy grid. If the parameter is set to CALDB, the file is read from the calibration database.

onaxiscfile [filename]
Name of file and extension that contains the on-axis telescope effective area, appropriate for the instrument, pre-calculated on a coarse energy grid. If the parameter is set to CALDB, the file is read from the calibration database.

outfile [filename]
Name of the output ARF or RSP file.

regmode [string RADEC/DET] Type of coordinate system associated with the region file ('regionfile'), RADEC or DET.

regionfile [filename] Name of the source region selection file, in SKY (RA/DEC units) or DET coordinates. The format is that of a standard SAO region file.

mirrorfile [filename]
Name of the telescope description file (TDF) and the extension (e.g., MIRROR) that holds the geometrical description of primary and secondary mirror foils. It is assumed that the name of the pre-collimator extension is COLLIMATOR. If the parameter is set to CALDB, the file is read from the calibration database.

obstructfile [filename]
The name of the telescope description file (TDF) and the extension (e.g. OBSTRUCT) that holds the geometrical description of the telescope support structures. If the parameter is set to CALDB, the file is read from the calibration database.

frontreffile [filename]
The name of the reflectivity file and the extension for the front-side reflectivity of the mirror foils. The extension also includes the thin surface film transmission. The names of the reflectivity and transmission columns are linked to groups of mirror foils that they apply to, by means of a column called FREFLECT in the TDF. Note that the second extension in the reflectivity file contains mass-absorption coefficients that are used by the raytracing code 'xrtraytrace' to calculate transmission probabilities of the "thick" materials (as opposed to thin films) in the telescope.

backreffile [filename]

The name of the reflectivity file and the extension for the backside reflectivity of the mirror foils.

pcolreffile [filename]
The name of the reflectivity file and the extension for the reflectivity of the pre-collimator blades/foils. The pre-collimator reflectivity is the same for frontside and backside reflection.

scatterfile [filename]
The name of the file containing the scattering angle probability distributions for the direction of reflected rays relative to the specular direction. The file contains data for the front-side of mirror foils, the back-side of mirror foils, and for the pre-collimator blades. In general, foils in different physical regions of the telescope can have different scattering distributions. The column names are referenced in the SCATTER column in the TDF.

numphoton [integer]
The value of the parameter 'numphoton' corresponds roughly to the number of raytracing photons allocated to each attitude histogram bin (in the exposure map file), per energy grid point. If any attitude bin has a time interval significantly larger than the average time interval, 'aharfgen' allocates additional raytracing photons to the that attitude position.

(minphoton = 100) [integer]
The minimum number of photons that successfully reach the focal-plane, per raytracing energy grid point, that is acceptable to make a viable ARF. The number of focal-plane photons that contribute to the ARF must exceed minphoton for every energy, otherwise the program aborts.

sourcetype [string]
Method for treatment of the spatial distribution of the X-ray source. (1) Point: Point source at infinity. Photons arrive at random points on the active region of the telescope aperture. (2) Flatcircle: Extended source at infinity that has a spatial distribution that has uniform flux over a circular region (zero outside of the circle). (3) Betamodel: Extended source at infinity that has a spatial distribution described by the "beta model" (see 'betapars' parameter). (4) Image: A FITS image file (in RA/DEC coordinates) is used to create a photon event list using the simulation code 'heasim'. This event list is subsequently input into the raytracing code by 'aharfgen'. The input image file could be made from data from a different mission, or from a model. The name of the image file is given by the parameter 'imgfile'..

(betapars = "0.50 0.60 5.0") [string]
Parameters of the beta model if 'sourcetype' is set to 'betamodel' as follows: (1) beta model core radius in arcmin, (2) the index "beta" of the beta model, (3) the maximum radius (in arcmin) of the source spatial distribution.

(flatradius = 10.0) [real]
The radius (in arcmin), of the extended source for the flat spatial distribution option, 'sourcetype=flatcircle'.

imgfile [filename]
Name of the input image file to be used for raytracing if the input parameter 'sourcetype=image'. The image should be in the primary extension and coordinate system should be SKY (RA/DEC units). The minimal mandatory keywords are the standard ones describing the "X" and "Y" grids: CRPIX1, CRVAL1, CDELT1, CUNIT1, CRPIX2, CRVAL2, CDELT2, CUNIT2.

auxtransfile [filename]
Name of the input auxiliary transmission file. This file is used to apply additional transmission that is not accounted in the telescope calibration files used by the raytracing. This valid for HXI, SXI and SXS. By default is set to NONE. To apply the auxiliary transmission users has either to input a file or set to CALDB.

(rmfthresh = 1.0e-10) [real]
This parameter is only valid for the HXI not for the SXI or SXS. Define the lower threshold for writing non-zero matrix elements in the HXI output RSP file. If the matrix elements value is less than the value of rmfthresh then that matrix element is et to 0. It is not recommended to increase the value rmfthresh above the default value as it may reduce the compressibility of the output, making the matrix too large for the software to handle.

(polydeg = "DEFAULT") [string]
Polydeg defines the polynomial order for the fitting. The allowed values are: DEFAULT and 1 to 5 for the HXI and 1 to 10 for the SXS and SXI. For the HXI the DEFAULT value of polydeg is set to 5. For the SXI and SXS, the DEFAULT value is set to the order tested internally for fitting stability.

(seed = 29075) [integer]
Random number seed. If the value is zero then the code uses the system time to generate a random number.

(cleanup = no) [boolean]

Determines whether to delete temporary files (yes/[no]).

[ccldhm]

EXAMPLES
1. Make an ARF file for a point source in the SXI given the RA/DEC region selection file src_sky.reg, and an exposure map file src_expmap.fits made by the tool 'ahexpmap'. An example of the contents of the region file src_sky.reg is: # Region file format: DS9 version 4.1
global color=green dashlist=8 3 width=1 font="helvetica 10 normal" select=1 high lite=1 dash=0 fixed=0 edit=1 move=1 delete=1\
 include=1 source=1 wcs=wcs fk5 circle(140.0,35.0,150.0")

In this example the source RA and DEC coordinates are 140 and 35 degrees respectively, and the extraction region is a circle centered on the source, with a radius of 150 arcsec. The parameters 'source_ra' and 'source_dec' that should be entered are equal to the values of the corresponding source coordinates in the region file. For the SXS or SXI, 'ahsxtarfgen' (called by aharfgen) only uses the energy grid in 'rmffile' and not the actual matrix data, so the particular response matrix (RMF) specified for 'rmffile' is not important. The ARF made by 'ahsxtarfgen' is calculated on the exact energy grid in 'rmffile'.

aharfgen xrtevtfile=src_1_raytrace.fits source_ra=140.0 source_dec=35.0 telescop=HITOMI instrume=SXI teldeffile=CALDB \
 (ah_sxi_teldef_20140101v001.fits) emapfile=src_exmpmap.fits qefile=CALDB (ah_sxi_quanteff_20140101v001.fits) \
contamifile=CALDB (ah_sxi_contami_20140101v001.fits) rmffile=sxi.rmf erange="0.4 15.0 0.4 15.0" onaxiscfile=CALDB \
(ah_sxi_telarea_20140101v001.fits[EFFAREACRS]) onaxisffile=CALDB (ah_sxi_telarea_20140101v001.fits[EFFAREAFNE]) \
outfile=src_1_arf.fits regmode=SKY regionfile=src_sky.reg mirrorfile=CALDB (ah_sxi_mirror_20140101v001.fits[MIRROR]) \
obstructfile=CALDB (ah_sxi_mirror_20140101v001.fits[OBSTRUCT]) \
frontreffile=CALDB (ah_sxi_reftrans_20140101v001.fits[AH_SXT_FRONT])\
backreffile=CALDB (ah_sxi_reftrans_20140101v001.fits[REFPROBBACK])\
pcolreffile=CALDB (ah_sxi_reftrans_20140101v001.fits[REFPROBPCOL]) \
scatterfile=CALDB (ah_sxi_scatter_20140101v001.fits) numphoton=10000 sourcetype=point

The example run produces the output following files: (1) src_1_arf.fits; (2) aharfgen_region.lis (contains the names of region files made in detector coordinates: src_1_arf.arfregion0.reg to src_1_arf.arfregionN.reg where N is the number of attitude histogram bins in the exposure map file, extension 1; (3) src_1_raytrace.fits.

KNOWN BUGS
If 'aharfgen' is run with 'sourcetype=image', the number of photons generated from the image in the initial event list should be equal to the input parameter 'numphoton', but this is not the case. A way to mitigate the effects of this bug is to run 'aharfgen' with the intended value of 'numphoton' and note in the screen output the number of actual source events generated by the tool 'heasim.' Then interrupt 'aharfgen' and run it again with a new value of 'numphoton' set to the original value squared divided by the actual number of events generated by heasim.

NAME

ahbackscal -- Correct BACKSCAL keyword in a spectrum extracted with XSELECT.

USAGE

ahbackscal infile regfile expfile

DESCRIPTION

'ahbackscal' corrects the the BACKSCAL keyword in a spectrum file that has been extracted from XSELECT, using as input the extraction region file and an exposure map, which must be defined in the same coordinate system. The BACKSCAL keyword is used by XSPEC to properly scale the background spectrum before subtracting it from the source spectrum, since the extraction regions are usually different between source and background. XSELECT sets BACKSCAL according to the fraction of pixels contained in the extraction region compared to the number of pixels defined by the TLMIN and TLMAX of the event list coordinate columns. If this region contains bad pixels, bad columns, partially exposed pixels, regions outside of the field-of-view, the BACKSCAL keyword will be incorrect.

The task works in the following way:

1. Mask the exposure map with the region file so that parts outside the region are set to zero.

2. Sum up the pixel values in the good region.
3. Normalize this sum by the maximum value in the exposure map ('norm=MAX'), by the spectrum file exposure time keyword ('norm=EXPOSURE'), or by some specified numerical value ('norm=value'). This third option makes the tool mission-independent, because some "exposure maps" are already normalized to unity.
4. Divide this value by the number of pixels in the coordinate system, since this is how BACKSCAL is defined.

In general, 'norm=MAX' should be used to properly scale the number of good pixels in the exposure map. Due to spacecraft attitude motion and therefore pixels with partial exposure in 'expfile', the total number of good pixels will not necessarily be an integer, and 'ahbackscal' takes this correctly in to account. It is important to note that 'ahbackscal' only corrects the scaling used to subtract the background spectrum in XSPEC; spectral modeling already accounts for all pixels that are bad, partially exposed, or out of the field-of-view by using properly constructed response files (RMF and ARF). There is no output file, instead the BACKSCAL keyword is updated in the input spectrum file.

This tool is designed to be mission-independent, and will work for any PHA spectrum extracted from XSELECT as long as the SAO region file 'regfile' and exposure map image 'expfile' are specified in the same coordinates (e.g. DET or SKY).

PARAMETERS

infile [filename]
Name of input spectrum (PHA) file. The BACKSCAL keyword is updated in this file.

regfile [filename]
Name of input region file. The format is that of a standard SAO region file, and the coordinate system should be the same as that of 'expfile'.

expfile [filename]
Name of input exposure map file. The exposure map my be an image in the primary extension of this file, in the same coordinate system used by 'regfile'.

(norm = MAX) [string]
Normalization value for exposure map file. This value is used to scale the summed pixel values in 'expfile' within the region defined by 'regfile'. If set to MAX, the maximum pixel value is used. If set to EXPOSURE, the value of the EXPOSURE keyword in the first extension of 'expfile' is used. If set to a numerical value (can be floating point), that value is used.

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES

Correct BACKSCAL in Hitomi SXI source and background spectra, using the maximum value in the exposure map to normalize (the default behavior). Note that both use the same exposure map file.

ahbackscal infile=ah100050010sxi_p0100004b0_cl_src.pi regfile=sxi_src_wcs.reg expfile=ah100050020sxi_p0100004b0_cl.expo
ahbackscal infile=ah100050010sxi_p0100004b0_cl_bkg.pi regfile=sxi_bkg_wcs.reg expfile=ah100050020sxi_p0100004b0_cl.expo

Correct BACKSCAL in a Hitomi SXI spectrum, using the EXPOSURE keyword in the exposure map first extension to normalize.

ahbackscal infile=ah100050010sxi_p0100004b0_cl_src.pi regfile=sxi_src_wcs.reg expfile=ah100050020sxi_p0100004b0_cl.expo
norm=EXPOSURE

Correct BACKSCAL in a Suzaku XIS spectrum, specifying a value to use as the exposure map normalization. This should be the maximum expected value in the exposure map, usually the exposure time in seconds.

ahbackscal infile=xis_src.pha regfile=xis_src_wcs.reg expfile=xis_src.expo
norm=15549.23

SEE ALSO

xselect

LAST MODIFIED

December 2016

NAME ahcalcl32ti -- Compute L32TI from S_TIME

USAGE ahcalcl32ti infile outfile timfile

DESCRIPTION
This task computes and fills the L32TI column using the S_TIME column and the TIME_PACKETS extension of the input TIM file. The tool interpolates on the L32TI vs S_TIME data in the TIME_PACKETS extension. No other columns or keywords are modified.

PARAMETERS
infile [filename]
Name of the input FITS file with an S_TIME column.

outfile [filename]
Name of the output file which is just a copy of the input file with the L32TI column filled. No other columns or keywords are modified.

timfile [filename]
Name of the input TIM file, containing the extension TIME_PACKETS.

(l32ticol = L32TI) [string]
Name of the output L32TI column to fill. If a column with this name is already present in the input file, the tool stops with an error.

(interp = twopoint) [string]
Method of interpolation (nearest, twopoint, fourpoint)

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

1. Recompute the L32TI column from S_TIME and replace the existing values.
ahcalcl32ti in.fits out.fits tim.fits

NAME ahcalctime - Rerun the timing tools on an output directory from the pipeline

USAGE ahcalctime indir outdir

DESCRIPTION
'ahcalctime' is a script that recalculates the TIME column in the event files and the housekeeping (HK) files. As input, the script expects a directory structure as defined for a sequence downloaded from the archive.
The data from the 'indir' directory are copied to the 'outdir'. The directory structure inside indir is the same of outdir. Only the files and directories necessary to the tools are copied over. For example, the event_uf directory is copied, but not the event_cl.

To recalculate time the script runs the following tools:
Runs ahmktim on the tim file, after making a copy of the original tim file.
Runs ahtime on the general HK file
Runs sxssamcnt on the SXS HK file
Runs sxssamcnt on each SXS unfiltered event file
Runs sxssamcnt on the SXS lost GTI file
Runs ahtime on the SXS lost GTI file
Runs ahtime on each instrument HK file
Runs ahtime on each instrument unfiltered event file

INPUT

The input directory 'indir' must have the following directory structure:
  indir/auxil/general HK file, tim file
  indir/[inst]/hk/
  indir/[inst]/event_uf/*uf.evt
  indir/sxs/event_uf/*el.gti
This is the same structure as is output by the pipeline that is placed in the archive.

OUTPUT

The following directories and files are copied from the 'indir' to the 'outdir', and the new time is calculated only for these files:
  indir/auxil/*
  indir/[inst]/hk/*
  indir/[inst]/event_uf/*uf.evt
  indir/sxs/event_uf/*el.gti

If 'clobber=yes' and the 'outdir' exists, the 'outdir' is removed completely before beginning,
 and a new copy is made from 'indir'. If 'clobber=no' and the 'outdir' exists, the script exits with a failure. All of the individual tools operate as though 'clobber=yes'.
 The tools called within ahcalctime create and overwrite, for example, the existing (in the 'outdir') instrument hk file from ahtime. The files in 'indir' are untouched.
 The original tim file is copied to a [base]_orig.tim, and the new tim file only keeps the primary HDU and first extension in order to recreate the second extension. When calling the timing tools, the script creates intermediate files, then either copies them or moves them to their final destination, based on the 'cleanup' parameter.

PARAMETERS
indir [string]
Input directory

outdir [string]
Output directory.

(verify_input = no) [boolean]
Verify the input files with ftverify.

(sorttime = yes) [boolean]
Sort the output time column after calling ahtime, on the column specified in the 'timecol' parameter. All extensions are sorted on this column, if that extension contains the column and has a nonzero number of rows.

(frqtemfile = CALDB) [filename]
Used in ahmktim: the freq vs. temp file (or CALDB)

(sxs_coeftime = CALDB) [filename]
Used in sxssamcnt: Input file with arrival time coefficients (or CALDB)

(coldeffile = CALDB) [filename]
Used in ahtime: Input file with column names for the time corrections (or CALDB)

(delayfile = CALDB) [filename]
Used in ahtime: Input instrument delay file (or CALDB)

(offsetfile = CALDB) [filename]
Used in ahtime: Input file with the CAMS offset corrections (or CALDB)

(leapsecfile = REFDATA) [filename]
Used in ahmktim, ahtime: Input leap second file (or CALDB, [REFDATA])

(timext = TIM_LOOKUP) [string]
Used in ahmktim: Output TIM extension created by the ahmktim tool.

(gaptime = 2.0) [real]

Used in ahmktim: Minimum time [s] to define a gap

(timecol = TIME) [string]
Used in ahtime: Output time column

(gticolumns = START) [string]
Used in ahtime: Output GTI column names

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
ahcalctime indir outdir
The script copies the necessary files from indir to outdir, then recalculates the time columns on the files in outdir.

NAME ahexpmap - Generates an exposure map for HXI, SXI, and SXS, or a flat field image for SXI and SXS

USAGE ahexpmap ehkfile gtifile instrume badimgfile pixgtifile outfile outmaptype delta numphi

DESCRIPTION
'ahexpmap' creates either an exposure map (for HXI, SXI, and SXS) containing an integrated exposure time for each pixel (with the parameter setting 'outmaptype=EXPOSURE') or a flatfield image (for SXI and SXS) where efficiencies, such as telescope vignetting and detector quantum efficiency, are multiplied (with the parameter setting 'outmaptype'=EFFICIENCY'). The EXPOSURE output is an input to the 'aharfgen' task, whereas the EFFICIENCY output may be directly used for exposure-correction in imaging analysis.
For the both 'outmaptype' modes, the task requires an EHK file ('ehkfile') and a GTI file ('gtifile') as input. The EHK file contains orbital and attitude information. The GTI file is the same used to screen the event file (typically the second extension of the event data). In addition, two files, bad image file ('badimgfile') and pixel GTI file ('pixgtifile') might be input optionally. The task reads the input EHK file and calculates a residual (off-axis) of the satellite attitude from the mean optical axis for each time stamp. Then it determines a time interval for each off-axis wedge of which size may be set by the parameters 'delta' and 'numphi'. When 'delta=A' and 'numphi=B' (A is a real number and B an integer), the off-axis wedges are populated as follows: First, the sky region is divided into a central circle and outer annuli with the radii of A arcmin. Then, the innermost annulus (just out of the central circle) is divided into B wedges. Similarly the 'n'-th annulus is divided into B*'n' wedges.
If 'outmaptype=EXPOSURE' and 'instrume=SXI or SXS', then the output file contains an exposure map image in the primary extension (where each pixel contains net exposure time in second) and an off-axis histogram table in the first extension. This table contains information such as the average attitude and exposure time of each off-axis wedge. The following extensions (from the second to the last) are GTI table for each off-axis. The output exposure map may be in either DET, FOC, or SKY coordinate set by the parameter 'stopsys'. If 'stopsys=SKY', the task uses the attitude information in the EHK file for the coordinate transformation.
If 'outmaptype=EXPOSURE' and 'instrume=HXI1 or HXI2', then the output file contains only the off-axis histogram table and GTI tables, and no image is created. The task 'hxirspeffimg' needs to be ran to generate an HXI flat field.
If 'outmaptype=EFFICIENCY', the output file contains a so-called flat field image in the primary extension, in which each pixel has a dimensionless efficiency value between 0 and 1. In this mode, the task reads efficiency CALDB files set by the parameters 'qefile', 'contamifile', 'vigfile', 'obffile', 'fwfile', and 'gvfile' (the last three are for only SXS), and multiplies all the efficiencies at the energy or energy range specified by the parameters 'specmode' and 'energy' (see below) to calculate the flat field image. Each of the efficiency components may be turned off by setting the parameter to NONE (e.g., 'vigfile=NONE' means "vignetting is not taken into account"). There are two ways to set the energy or energy range for which the efficiencies are used. When 'specmode=MONO', the parameter 'energy' should be a single value, and the efficiency at this energy is read from the CALDB files. When 'specmode=SPEC', a spectrum file is also required, and 'energy' indicates the minimum and maximum of the desired energy range (two values separated by a comma, e.g., "1.0,5.0"). The name and format of the spectrum file must be specified by the parameters 'specfile' and 'specform', respectively.
The optional bad image file ('badimgfile') is a FITS image containing the following flag values for each pixel: good pixels (0), calibration source regions (1), bad pixels and columns (2), and out of the detector or area discrimination (-1, null). For the SXI, this is the output of the 'sxiflagpix' task. If 'badimgfile=NONE', then the active pixel location is found from the input 'instmap' CALDB. However, the CALDB file does not contain the bad/hot pixel or calibration source location, and distinguish only inside or outside of the detector.
The optional pixel GTI file ('pixgtifile') contains the pixel location in the DET coordinate with START and STOP time for each dead time interval (e.g., the pixel is "bad" between START and STOP). This file is needed to account for the time-dependent bad pixels (e.g., flickering pixels for SXI).

In the default setting, the calibration source regions (for SXI) is masked out. However, these regions may be regarded as good pixels by setting 'maskcalsrc=no'.

PARAMETERS
ehkfile [filename]
Name of input EHK file.

gtifile [filename]
Name of input GTI file.

instrume [string]
Name of the instrument (SXS, SXI, HXI1, or HXI2).

badimgfile [filename]
Name of input bad pixel image file. The file should contain an image in the primary extension with the following flag values: good pixels (0), calibration source region (1), bad pixels and columns (2), and out of the detector or area discrimination (-1, null). For the SXI, this is simply the output of the sxiflagpix routine ('outbadimg').

pixgtifile [filename]
Name of input pixel GTI list. The file contains a bin table with columns of START, STOP, DETX, and DETY, indicating the duration and location of each partial bad pixel.

outfile [filename]
Name of output file.

outmaptype [string]
Type of the output map (EXPOSURE or EFFICIENCY).

delta [real]
Size in arcmin of each off-axis annulus grid (and on-axis circle) in the output histogram.

numphi [integer]
Number of azimuth (phi) bins in the first off-axis annulus at (delta <= theta < 2 * delta). A n-th off-axis annulus has azimuth bins of numphi * n.

(stopsys = SKY) [string]
Output coordinate system (DET, FOC, or SKY).

(instmap = CALDB) [filename]
Name of input instrument map file. If the parameter is set to CALDB, the file is read from the calibration database. This file is required for the task to run.

(qefile = CALDB) [filename]
Name of input quantum efficiency file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE.

(contamifile = CALDB) [filename]
Name of input contamination file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE.

(vigfile = CALDB) [filename]
Name of input vignetting coefficient file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE.

(obffile = CALDB) [filename]
Name of input optical blocking file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE or 'instrume' is not SXS.

(fwfile = CALDB) [filename]
Name of input filter wheel file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE or 'instrume' is not SXS.

(gvfile = CALDB) [filename]
Name of input gate valve file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE or 'instrume' is not SXS.

(maskcalsrc = yes) [boolean]
If this parameter is set to 'yes', calibration source regions are regarded as bad pixels and excluded from the output exposure map. This parameter is ignored when 'badimgfile' is set to NONE.

(fwtype = OPEN) [string]
Filter wheel type for the SXS (OPEN, FE55, BE, ND, or POLY) This parameter is ignored when 'outmaptype' is set to EXPOSURE or 'instrume' is not SXS.

(specmode = MONO) [string]
Type of input energy: monochrome energy (MONO) or spectrum (SPEC).

(specfile) [filename]
Name of input spectrum file. This parameter is ignored if 'specmode' is set to MONO.

(specform = FITS) [string]
Format of the input spectrum file (FITS or ASCII). This parameter is ignored if 'specmode' is set to MONO.

(energy = 6.0) [string]
An energy value or an energy range for the spectrum from which a flux-weighted efficiency is calculated. When 'specmode' is set to MONO, this parameter should be a single value (e.g., '6.0') When 'specmode' is set to SPEC, this parameter should be two values separated by a comma (e.g., '1.0,5.0') to specify the input energy range.

(evperchan = DEFAULT) [string]
Energy (in eV) per channel of the input spectrum file. This parameter is ignored if 'specmode' is set to MONO. If this parameter is set to DEFAULT (as default) and 'instrume' is set to SXI, a 6.0 eV/channel is assumed so that the value matches a standard PHA file of SXI. Similarly, if this parameter is set to DEFAULT (as default) and 'instrume' is set to SXS, a 0.5 eV/channel is assumed.

(abund = "1.0") [string]
Relative abundances of contaminants. The abundances of all of the contaminant materials in the calibration file are multiplied by this number.

(cols = "0.0") [string]
Additional column densities for contaminants (1E18 cm^-2). The column densities of all of the contaminant materials in the calibration file are modified by adding the value of this parameter.

(covfac = "1.0") [string]
Partial covering factors for contaminant materials. The partial covering factors of all of the contaminant materials in the calibration file are multiplied by this number.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

1. Create exposure map for the SXS with off-axis bin size of 0.5 arcmin. No pixel GTI file is input. Exclude calibration source as bad pixel.
ahexpmap ehkfile=ah.ehk gtifile=gti.fits instrume=SXS badimgfile=NONE pixgtifile=NONE outfile=outexpmap.fits \
outmaptype=EXPOSURE delta=0.5 numphi=4 maskcalsrc=yes

2. Create efficiency map for the SXI with off-axis bin size of 1.0 arcmin. Flickering pixel file is input as pixel GTI. Include calibration source regions as good pixel.
ahexpmap ehkfile=ah.ehk gtifile=gti.fits instrume=SXI badimgfile=badpiximage.fits pixgtifile=flickpix.fits outfile=outflatfield.fits \
outmaptype=EFFICIENCY delta=1.0 numphi=4 maskcalsrc=no

NAME ahfilecaldb - create a FITS file from a set of text files

USAGE ahfilecaldb.pl infile outfile

DESCRIPTION
This script provides a tool that creates a FITS file with one or more extensions from a set of text files containing the data and table definition. The FITS file may have multiple extensions. This script used the ftool ftcreate to create the output FITS file.

PARAMETERS
infile [file]
Input ASCII text file listing the filenames of other set(s) of ASCII files that populates the FITS file. Each row contains the files that go into one extension. Each row should consist of three strings, specifying (in order) the data filename, the column definition filename, and the header filename, delimited by spaces: data.dat column_def.txt header.txt. The files must be in the same format as specified in the help for ftcreate.

The column definition file specifies the commodity of each column in the data file, and its data type:
```
PAR  2A
COEF0  1E
COEF1  1E
COEF2  1E
COEF3  1E
```

For each row in the column definition file, there corresponds one column in the data file:
```
P0  1.6419760E+00 6.9057592E-04  -2.8387581E-03  -1.5602625E-05
P1  2.7251086E+00 4.8774816E-03  -2.9882593E-03  -7.2517512E-05
P2  1.3079076E+00 1.1485046E-03  -2.9651825E-03 1.1986124E-04
P3  3.2381654E-01 3.2554791E-04  -2.2113894E-03 8.9718503E-07
```

The header file contains the keyword name, the keyword value, and the keyword description:
```
EXTNAME 'PSF_COEF'   / Name of this binary table extension
TELESCOP  'SWIFT '   / Telescope (mission) name
INSTRUME  'XRT '   / Instrument name
```

outfile [file]
The name of the output fits file that contains all the extensions created from the data and definitions in the infile.

[ccldhm]

EXAMPLES
Create a CALDB file with PSF coefficients. The file has two extensions.
ahfilecaldb.pl in.txt out.fits
The file in.txt contains:
input/data_psf.txt input/dd_psf.txt input/head_psf.txt
input/data_2.txt input/dd_psf.txt input/head_psf.txt

NAME ahfilter -- generate an EHK file and a MKF filter file

USAGE ahfilter mkconf attitude orbit reference

DESCRIPTION
'ahfilter' script runs the 'ahmkehk' task to generate an expanded housekeeping file using orbit and attitude files for a given observation, and then runs 'makefilter' to generate an MKF filter file from an input configuration file. This MKF file combines information from various instrument, general housekeeping and auxiliary files for a given observation into a common time order datafile. The EHK and MKF files are used to filter the science data

'ahfilter' takes as input a configuration file that is either an ASCII file or a FITS file, possibly stored in CALDB. The configuration file contains the filename, extension, and column names of the parameters that need to be sampled as well as other characteristics; see the help for 'makefilter' for details on the configuration file format requirements. The tool also requires an observation-specific attitude file ('attfile') and orbit file ('orbfile') that describe the location and pointing of the telescope and are used as input to 'ahmkehk'. An optional time reference file ('reference') and additional CALDB files describing the orbit ('cor2file', 'cor3file', 'saafile') and time sampling of the EHK output file ('tsart', 'tstop', 'bintime', 'textend') are also passed to 'ahmkehk'. See the help for 'ahmkehk' for details.

The tool outputs an MKF file produced by 'makefilter', which can be further used with 'maketime' to create good time intervals (GTI) based on some filtering criteria. See the help for those tools for more information.

PARAMETERS
mkfconf [filename]
Input makefilter configuration file (or CALDB).

attfile [filename]
Input Hitomi attitude file.

orbfile [filename]
Input Hitomi orbit file, assumed to have the same format as Suzaku version 2 orbit files.

reference [filename]
Input time reference file (or NONE to use the attitude file) if 'tstart' or 'tstop' are set to 0.

(teldeffile = CALDB) [file]
Name of the input TELDEF file providing the coordinate systems of the SXI instrument, which is used fiducially to obtain sky coordinates (or CALDB).

(leapsecfile = REFDATA) [file]
Input leap second file (or CALDB, [REFDATA]).

(optaxis = 1215.5,1215.5,1215.5,1215.5,1215.5,1215.5,1215.5,1215.5) [string]
X and Y optical axis coordinates in FOC coordinates for HXI1, HXI2, SGD1, SGD2, SXI, and SXS instruments, respectively.

(cor2file = CALDB) [file]
Input cut-off rigidity Suzaku file (or CALDB).

(cor3file = CALDB) [file]
Input cut-off rigidity IGRF 2016 file (or CALDB).

(saafile = CALDB) [file]
Input SAA vertices for each instrument (or CALDB).

outehkfile [file]
Name of output EHK file.

outmkffile [file]
Name of output makefilter (MKF) file.

(attext = ATTITUDE) [string]
Name of the extension in the attitude file ('attfile') with pointing information.

(attform = EULER) [string]
Format of the input attitude column in the 'attext' extension of 'attfile'. Must be EULER or QUAT.

(attcol = EULER) [string]
Name of the input attitude column in the 'attext' extension of 'attfile'.

(orbext = ORBIT) [string]
Name of the extension in the orbit file ('orbfile').

(orbform = KEPLERIAN) [string]
Format of the input orbit velocity column in the 'orbext' extension of 'orbfile'. Must be VECTOR, COMPONENTS, or KEPLERIAN.

(orbcol = A,E,I,AN,AP,MA) [string]
Name of the input orbit columns in the 'orbext' extension of 'orbfile'.

(timecol = TIME) [string]
Name of the column containing mission time in the reference file.

(tstart = 0.0) [real]

(tstop = 0.0) [real]
End of mission time interval of the output file [s]. If either tstart or tstop is 0.0, the time interval is set equal to the time range of the reference file if there is one, or else to that of the attitude file.

(bintime = 1.0) [real]
Time sampling interval of the output file [s].

(textend = 0.0) [real]
Time added to the beginning and end of the output file time interval, preceding tstart and following tstop [s].

(infileroot =) [string]
Prefix of the input HK files for applying 'mkconf' file.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Run ahfilter using the configuration file in CALDB, and the attitude, orbit, and housekeeping files corresponding to sequence ah000507024.
```
ahfilter mkfconf=ah_gen_mkfconf_20131001v004.fits attfile=ah000507024.att.gz orbfile=ah000507024.orb.gz reference=NONE \
outehkfile=ahfilttest1.ehk outmkffile=ahfilttest2.mkf infileroot="ah000507024" clobber=yes
```

NAME ahgainfit -- Calculate the time-dependent energy gain corrections using known calibration lines

USAGE ahgainfit infile outfile

DESCRIPTION
'ahgainfit' calculates time-dependent energy gain corrections by comparing the theoretical and observed energies of a calibration line or line complex. For each run of the task, only one line may be specified to calculate the gain correction (see parameter 'linetocorrect'). 'ahgainfit' is a general task used directly in the scripts 'sxigainfit', 'hxigainfit', 'sgdgainfit' for the SXI HXI, and SGD respectively. For the SXS, the 'sxsgain' tasks uses the same core fitting function of 'ahgainfit'.

'ahgainfit' takes as input an event file with time and energy columns, and requires that the events are time ordered. The task accumulates spectra from events that are consecutive in time, with energy centered on the calibration feature and compares each spectrum from the events with a theoretical model of the calibration feature profile. The calibration feature used by 'ahgainfit' is specified by the parameter 'linetocorrect' as a string, and the names and energies of the features are specified in a calibration file (parameter 'linefitfile'). The calibration feature may be composed of many atomic or nuclear line components that are listed in the calibration file.

The energy range for the spectra constructed from the event file as well as from the theoretical profile, may be specified in two different ways. (1) The default energy range for the spectra is the smallest and largest energies of the line components, and expanded with the 'extraspread' parameter, i.e. [E_min - extraspread : E_max + extraspread]. It is recommended to set 'extraspread' larger than the sum of the natural width of the calibration feature, the value of the 'broadening' parameter, and the magnitude of the expected energy shift. (2) Alternatively, the energy range may be specified by setting the 'startenergy' and 'stopenergy' parameters. If these parameters are non-negative 'ahgainfit' uses their values to accumulate the spectra, instead of the range derived using the 'extraspread' setting. The energy column used to accumulate the spectra is specified by the 'energycol' parameter. The energy column is expected to be in units of channel, where the eV per channel is set by the 'evchannel' parameter. The spectra are binned according to the 'binwidth' parameter, where the 'binwidth' is in units of 'energycol' channels. The theoretical profile is constructed on a mesh defined by this energy range and 'binwidth', where each calibration line is assumed to be Lorentzian. The profile may be convolved with a Gaussian having the FWHM given by the 'broadening' parameter.

The number of events in each spectrum is defined by the 'numevent' and 'minevent' parameters. The task accumulates spectra with a number of events between 'minevent' and 'numevent'. However, if a spectrum has fewer than 'minevent' points, then it is combined with the previous spectrum if possible. Therefore all spectra have a size between 'minevent' and ('numevent+minevent-1'). To avoid having spectra accumulated over large gaps in time, the group of points in the spectrum is truncated when the time interval between consecutive events is greater than the 'gapdt' parameter. Adjacent spectra in time may share a percentage of their points based on the 'grpoverlap'

parameter that may vary between 0 and 100. If 'grpoverlap' is set to 0, the consecutive spectra share no points in common; if set to 100 they share all points in common but one.

The spectra events may be simultaneously collected based on the value of a column present in the event file, given by the 'splitcol' parameter. This option may be used, for example, to find the gain correction for each layer of the HXI detector ('splitcol=LAYER'). If a GTI file is specified by the 'gtifile' parameter, events outside of these GTI intervals are excluded. Spectra are not accumulated across GTI intervals unless the 'spangti' parameter is set to 'yes'.

For each accumulated spectrum, 'ahgainfit' fits the theoretical profile to the data and also derives binned and unbinned averages. A least-squares method is used in the fitting. The fitted parameters are energy shift, scaling factor, background (unless the 'background' parameter is set to NONE), and, optionally, convolution width if 'fitwidth=yes'. The background is fit with a constant value if 'background' is set to CONSTANT, and a power-law if set to SLOPE. The unbinned average energy (as specified by 'energycol') is the sum of the energies in the spectrum divided by the number of events in the spectrum. The binned average energy is the weighted average derived by summing, over bins in the spectrum, the product of the energy and number of events per bin, and then dividing by the total number of events in the spectrum. The fitted gain correction is computed from the fitted shift with respect to the theoretical line profile. The binned average gain correction is computed from the difference between the profile and spectrum averages.

The default values for the parameters used in the fitting method ('minwidth0', 'maxitcycle', 'r2tol', 'searchstepshift', 'maxdshift', 'bisectolshift', 'searchstepwidth', 'maxdwidth', 'bisectolwidth', and 'minwidth') should not need to change since already optimized.

The output file has two extensions. One extension, GRID_PROFILE, contains the energies and amplitudes of the theoretical profile used in the fitting procedure, including any convolution from the 'broadening' parameter. The other extension, DRIFT_ENERGY, reports the fitting results for each spectra in the following columns: TIME (midpoint of the time interval over which the spectrum is collected), splitcol (value of splitcol for spectrum as given by the 'splitcol' parameter; this column is absent if 'splitcol=NONE'), COR_FIT (energy correction factor from spectrum fit), COR_AVE (energy correction factor from spectrum average), CHISQ (reduced chi-squared of the fit), AVGUNBIN (average energy of events in spectrum prior to binning), AVGBIN (weighted spectrum average energy), AVGFIT (average energy from fit), SHIFT (fitted energy shift), SCALE (fitted vertical scaling factor), BGRND (fitted background), WIDTH (if 'fitwidth=no', same as broadening parameter; if 'fitwidth=yes', fitted width), TELAPSE (difference between times of first and last event in spectrum), EXPOSURE (calculated using the GTI), NEVENT (total number of events collected for this spectrum), BINMESH (array containing the count spectrum energy bins), SPECTRUM (array containing the observed binned count spectrum), FITPROF (array containing theoretical profile with fitted parameters applied). If the 'calcerr' parameter is set to 'yes', one-sigma errors for the SHIFT and WIDTH are calculated. The errors are calculated with chi-squared and maximum-likelihood methods and output in the columns SIGSHCHI2, SIGWDCHI2, SIGWDLIKE and SIGWDLIKE respectively. If 'writeerrfunc' parameter is set, the chi-squared and likelihood calculated valued are output in the arrays SHCHI2, SHLIKE, WDCHI2 and WDLIKE. The numbers of values output in these arrays are specified in the 'nerrshift' and 'nerrwidth' parameters, respectively.

PARAMETERS

infile [filename]
Input event file name. The file must have the TIME column filled.

outfile [filename]
Output gain correction file name.

(linefitfile = CALDB) [filename]
Input CALDB file containing parameters of the Lorentzian components used to construct the theoretical line profile. If the parameter is set to CALDB, the file is read from the calibration database.

(linetocorrect = Mnka) [string]
Calibration line to use for the gain correction. The value must match an extension in linefitfile.

(energycol = UPI) [string]
Name of energy column to use in gain fitting.

(splitcol = PIXEL) [string]
Column name used to separate the data.

(numevent = 250) [integer]
Number of events collected for each spectrum used to calculate a single gain correction.

(minevent = 150) [integer]

Minimum number of events required for a spectrum. If the length of a group is less than minevent, those points are included with the previous group for processing, if possible.

(gtifile = NONE) [filename]
Input GTI file with time intervals to consider. Set to NONE to use entire input file.

(gapdt = -1) [real]
The upper limit to the time interval between two consecutive events in the same spectrum used in fitting. Two consecutive events separated in time by more than this amount are assigned to different groups. If gapdt=-1, no limit is imposed.

(grpoverlap = 0) [real]
The percentage overlap between adjacent groups. For grpoverlap=100 adjacent groups are shifted by one event, for grpoverlap=0 adjacent groups are independent and share no events.

(startenergy = -1.) [real]
Beginning of energy range in eV over which the spectra are collected. If startenergy is negative, the first energy is automatically determined by the smallest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(stopenergy = -1.) [real]
End of energy range in eV over which the spectra are collected. If stopenergy is negative, the final energy is automatically determined by the largest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(extraspread = 10.) [real]
Energy in eV by which the energy range is extended on either side beyond the smallest and largest energy in the linefitfile for the selected calibration feature. This parameter may be overridden by the startenergy and stopenergy parameters.

(evchannel = 1.) [real]
Conversion factor from channel number (for energycol) to energy [eV/chan].

(binwidth = 1.) [integer]
Energy bin width, in units of channels, to use when collecting spectra.

(broadening = 1.0) [real]
FWHM of the Gaussian in eV used to initially broaden the theoretical line profile. If fitwidth is set to no, the profile width is fixed at this value.

(gridprofile = no) [boolean]
If gridprofile is set to yes, only output the theoretical profile including any convolution due to the broadening parameter; no fitting is conducted (yes/[no]).

(fitwidth = no) [boolean]
If fitwidth is set to yes, then fit the width of each spectra in addition to the energy shift (yes/[no]).

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE).

(spangti = no) [boolean]
If spangti is set to yes, events in different intervals in gtifile may be collected in the same spectrum to be fit. If spangti is set to no, then groups of events used to construct the spectra must be from the same GTI. This parameter is ignored if gtifile is set to NONE (yes/[no]).

(avgwinrad = -1.) [real]
Radius of interval (in units of binwidth) used only to update the initial shift estimate prior to fitting. If avgwinrad is set to -1, this radius is automatically calculated based on the theoretical line profile and the broadening parameter. This is not used in calculating the average results.

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no]).

(writeerrfunc = no) [boolean]
Output the array of chi-squared and likelihood calculated for the SHIFT and WIDTH (yes/[no]).

(minwidth0 = 1.0) [real]
Smallest width, in units of binwidth, allowed as the initial value in width fitting. This parameter provides a lower limit to the initial estimate of the width as computed by the fitting algorithm. The value must be greater than zero.

(maxitcylce = 5) [integer]
Maximum number of fitting iterations.

(r2tol = 0.001) [real]
Convergence criterion on R-squared for least-squared fitting. Once R-squared changes by less than this amount between fitting iterations, the procedure is finished. This parameter should not normally need to be changed from the default value.

(searchstepshift = 2.) [real]
Step size, in units of binwidth, used when searching for best-fit energy shift in either direction from the initial shift estimate based on the spectrum average. The final shift is obtained using the bisection method (see bisectolshift).

(maxdshift = 5.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of energy shift. If no solutions are found within this deviation at smaller or larger shifts from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolshift = .001) [real]
When the bisection method determines the energy shift to within this amount in units of binwidth, the fitting procedure is completed.

(searchstepwidth = 5.) [real]
Step size, in units of binwidth, used when searching for best-fit convolution width in either direction from the initial width estimate based on the difference between the profile and spectrum statistical variances. The final width is obtained using the bisection method (see bisectolwidth).

(maxdwidth = 10.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of convolution width. If no solutions are found within this deviation at smaller or larger widths from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolwidth = .001) [real]
When the bisection method determines the convolution width to within this amount in units of binwidth, the fitting is procedure is completed.

(minwidth = .5) [real]
Since the least-squares fitting functional is undefined when the width is zero, one must define a minimum allowed fitted width. If the fitting routine attempts to fit a width smaller than this value (in units of binwidth), the fitting procedure fails for the spectrum.

(nerrshift = 100) [int]
Number of shift values in uncertainty calculations.

(nerrwidth = 100) [int]
Number of width values in uncertainty calculations.

(shifterrfac = 3.0) [real]
Factor for determining domain of shift uncertainty arrays.

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Compute the gain correction for an SXI event file using Mn K lines as the theoretical profile. The theoretical profile is convolved with a 12 eV Gaussian and gains are computed separately for each CCD_ID.
ahgainfit infile=event_in.fits outfile= drift_out.fits linetocorrect=Mnka energycol=PI splitcol=CCD_ID extraspread=20. \
evchannel=6. broadening=12. fitwidth=yes

2. Compute the gain correction for an HXI event file using Mn K lines as the theoretical profile. The theoretical profile is convolved with a 200 eV Gaussian and gains are computed separately for each SIDE.
ahgainfit infile=event_in.fits outfile=drift_out.fits linetocorrect=Mnka energycol=PI splitcol=SIDE extraspread=400. evchannel=100. \
broadening=200. fitwidth=yes

3. Compute the gain correction for an SGD event file using the Sn119 line as the theoretical profile. The theoretical profile is convolved with a 1500 eV Gaussian and gains are computed separately for each MATTYPE.
ahgainfit infile=event_in.fits outfile=drift_out.fits linetocorrect=Sn119 energycol=PI splitcol=MATTYPE extraspread=2500. \
evchannel=1750. broadening=1500. fitwidth=yes

NAME ahgetvector - Extract vectors from a FITS file and output into an ASCII file

USAGE ahgetvector infile outfile selmode xcol ycol row element

DESCRIPTION
'ahgetvector' script extracts vectors from a pair of columns in a FITS file table and formats the data as text in an ASCII file with two columns. This format conversion facilitates the plotting of selected elements in columns with multiple elements. The input FITS file extension must be in the form of a table where columns are either a single element array or a multi-element array. The task may output the elements in two columns with the same number of elements for a given row by setting 'selmode=2', selecting the row number using the 'row' parameter, and the names of the two columns using the 'xcol' and 'ycol' parameters. Alternatively, the task may output the values in single-element column and those for a particular element in a multi-element column for each row by setting 'selmode=1'. In this case the single-element column is specified by 'xcol' the multi-element column by 'ycol', and element index selected using the 'element' parameter. This element index must be within the valid range of vector elements for the 'ycol' column. For example, if the TFORM for 'ycol' is 3E, then 'element' must not be larger than 3. The file name and extension are determined by the 'infile' parameter. If no extension is specified, the first extension is used.

PARAMETERS
infile [filename]
Name of input FITS file.

outfile [filename]
Name of output ASCII file, or STDOUT to write to screen.

selmode [integer]
Select the data by either sub-element ('selmode=1') or by row ('selmode=2'). If selecting by sub-element, the parameter 'element' must be entered. If selecting by row, the parameter 'row' must be entered.

xcol [string]
Single column name to output as the first column in the output file.

ycol [string]
Single column name to output as the second column in the output file.

row [integer]
If the parameter 'selmode' is set to 2, this parameter specifies which row to retrieve. Must be a single row number, not a range of rows. This parameter is not used when 'selmode' is set to 1.

element [integer]
If the parameter 'selmode' is set to 1, this parameter specifies which element to retrieve. This must be within the valid range of vector elements for the 'ycol' column. This parameter is not used when 'selmode' is set to 2.

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

Extract vectors from a reflectivity probability file by element. Get the reflectivity for every energy, for the 2nd angle.
ahgetvector.pl infile=ah_sxt_xrtreftb.fits[AX_SXT_FRONT] outfile=out.txt selmode=1 xcol=energy ycol=refprob01 row=- element=2
Where the input file has 3 columns with the format:
Energy Angle RefProb01
keV rad
1E 2E 2E
And 4 rows under each column:
Energy Angle RefProb01
10 0.02 0.03 0.51 0.29
12 0.02 0.03 0.44 0.27
14 0.02 0.03 0.41 0.23
15 0.02 0.03 0.31 0.17
The output ASCII file out.txt is as follows:
energy refprob01
10 0.29
12 0.27
14 0.23
15 0.17

Extract vectors from a reflectivity probability file by row. Get the reflectivity for every angle, for the 3rd energy.
ahgetvector.pl infile=ah_sxt_xrtreftb.fits[AX_SXT_FRONT] outfile=out.txt selmode=2 xcol=angle ycol=refprob01 row=3 element=-
Where the input file has 3 columns with the format:
Energy Angle RefProb01
keV rad
1E 2E 2E
And 4 rows under each column:
Energy Angle RefProb01
10 0.02 0.03 0.51 0.29
12 0.02 0.03 0.44 0.27
14 0.02 0.03 0.41 0.23
15 0.02 0.03 0.31 0.17
The output ASCII file out.txt is as follows:
angle refprob01
0.02 0.41
0.03 0.23

NAME ahgtigen - create and/or merge GTI files

USAGE ahgtigen infile outfile gtifile gtiexpr mergegti

DESCRIPTION
'ahgtigen' has two primary functions. It can either create a GTI file using an expression operating on columns stored in a FITS file, or it can merge two or more GTI files.

The first function requires an input housekeeping (HK) (or similarly structured time-ordered fits file of quantities to use for filtering) file, or a text file containing a list of fits files with identical structure and extensions, specified by the 'infile' parameter (preceded by "@" if input is a list). In this case, the 'gtifile' parameter should be set to NONE, and the 'mergegti' parameter is ignored. The name of the TIME column in the input file(s) may be specified using the 'time' parameter.

Users may specify an additional expression via the 'gtiexpr' parameter or use a pre-existing CALDB filtering expression stored in the label file specified by the 'selectfile' parameter. The label file is a FITS file containing several expressions, each with a label that identifies the filtering details (the screening expression, and the type of file the expression is applied to) used to screen events and create the GTI. A user-input label file may also be used via the 'selectfile' parameter. A single label, or any combination of labels in the form of a comma-separated list may then be specified using the 'label' parameter file. In the latter case, the resulting GTI is created by conjoining the individual expressions using AND. The user may include an expression using the 'gtiexpr' parameter in combination with an expression (or expressions) from the label file. In this case the expressions are combined using AND. If multiple files are input using a list for 'infile', and/or there are multiple extension in the input file(s) the output gtifile have multiple extensions, each corresponding to a single extension of a single file. The names of the start and stop times in the output GTI files may be set using the 'outstart' and 'outstop' parameters may also be specified.

The second function of ahgtigen is to merge a list of GTI files (preceded by "@" if input is a list) specified by the parameter 'gtifile' using the mgtime ftool, according to the value of the 'mergegti' parameter (either AND or OR). In this case, the 'infile' and 'gtiexpr' parameters may be set to NONE; otherwise the output of the operation of the first function is merged with that of the second function. Users may update the timing keywords (EXPOSURE, TSTART, TSTOP, etc.) using the 'upkeyword' parameter, and other keywords may be copied from the first input file onto the final file using the 'cpkeyword' parameter. The mgtime parameters 'instarts' (name of the gti start times column), 'instops' (name of the gti stop times column) may also be specified.

PARAMETERS

infile [filename]
Input file with parameters for creating the GTI, or name of text file with list of files, preceded by "@" if creating a gti with multiple extensions from a list of files. Set to NONE if merging existing GTI files only.

outfile [filename]
Name of output GTI file.

gtifile [filename]
Input gti file, or name of text file with list of files, preceded by "@" if merging multiple input gti files. Set to NONE if creating gti file(s) from input files only.

gtiexpr [string]
User-input expression applied to input file used to create GTI. Set to NONE if merging gti only or using label file for selection only.

mergegti [string]
Merge GTI mode (OR or AND) used in combining GTI files. Required only if gtifile is not set to "NONE".

(cpkeyword = NONE) [string]
List of keywords to be copied from the first input file into the output gti file. May be set to NONE or ALL.

(upkeyword = yes) [boolean]
Update timing keywords in output file using the first input file ([yes]/no).

(leapsecfile = REFDATA) [filename]
Name of the file giving the table of leap seconds. If the parameter is set to CALDB, the file is read from the calibration database.

(instrume = NONE) [string]
Instrument used to search label file.

(selectfile = NONE) [string]
CALDB or user input label file with labels and expressions. May be set to NONE if merging gti, or using gtiexpr parameter.

(label = NONE) [string]
Labels to read from label file specified by 'selectfile'. To input multiple labels, use a comma-separated list.

(outexpr =) [string]
Final expression used to create GTI. Output parameter only.

(instarts = START) [string]
Name of input column containing GTI start times.

(instops = STOP) [string]
Name of input column containing GTI stop times.

(time = TIME) [string]
Name of column containing HK parameter times.

(outstart = START) [string]
Name of output column containing GTI start times.

(outstop = STOP) [string]
Name of output column containing GTI stop times.

(prefr = 0.) [real]
Pre-time interval factor. Must be within [0,1].

(postfr = 1.) [real]
Post-time interval factor. Must be within [0,1].

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no).

[ccldhm]

EXAMPLES
1. Run ahgtigen with the mkf file CALDB expression labeled by PIXELALL, creating an output GTI file.
ahgtigen infile=infile.mkf outfile=outfile_mkf.gti gtifile=NONE gtiexpr=NONE mergegti=AND selectfile=CALDB label=PIXELALL

NAME ahmkehk - Extract and calculate parameters from orbit and attitude (EHK)

USAGE ahmkehk attfile orbfile outfile

DESCRIPTION
'ahmkehk' reads satellite orbit and attitude files for a given observation, and generates an Expanded Housekeeping (EHK) file. The EHK file contains many parameters related to the attitude SAA, cut-off rigidity, and other orbital information, that are not supplied in any housekeeping or other auxiliary file. The content of the EHK is use to screen the data.

The time coverage of the output file is based on the 'tstart' and 'tstop' input parameters (seconds since the start of the mission). The input 'attfile' is used as the basis of the output times if 'tstart' or 'tstop' is 0 and 'reference'=NONE. These start and stop times can be extended using the 'textend' parameter, and the output columns are calculated by extrapolation if necessary. The 'bintime' parameter defines the time sampling of the output file. All times are in seconds.

The pointing is derived from the columns specified by 'attform' parameter, in the extension specified by the 'attcol' parameter, in the file specified by the 'attfile' parameter. The column must be in the format of either a three-component Euler angle array ('attform'=EULER) or four-component quaternion ('attform'=QUAT).

The orbit position is computed from the orbital elements stored in the columns in the extension given by the 'orbext' parameter in the file given by the 'orbfile' parameter. If the 'orbform' parameter is set to KEPLERIAN, these columns are A (semi-major axis), E (eccentricity), I (inclination), AN (ascending node right ascension), AP (ascending node angle), and MA (mean anomaly) unless overridden by the 'orbcol' parameter. If 'orbform' is set to COMPONENTS, the individual components of the orbital velocity are extracted from the VX, VY, and VZ columns unless overridden by the 'orbcol' parameter. If 'orbform' is set to VECTOR, these velocity components are read from a single vector column specified by 'orbcol'.

The file corresponding to the 'teldeffile' and the values given by the 'optaxis' parameter contain the information needed to transform between coordinates related to pointing, Hitomi focal plane, and sky coordinate systems. The TELDEF file set by 'teldeffile' is that for a fiducial instrument defined as the SXI, and the 'optaxis' parameter is a comma-delimited list of Hitomi optical axis FOC coordinates for HXI1 (FOCX), HXI1 (FOCY), HXI2 (FOCX), HXI2 (FOCY), SXI (FOCX), SXI (FOCY), SXS (FOCX), and SXS (FOCY), respectively.

'ahmkehk' creates an output file that includes three columns that store values of the cutoff rigidity (COR, COR2, COR3). The rigidity tables (in the form of geographic longitude/latitude map images) that provide data for COR2 and COR3 are specified by the 'cor2file' and 'cor3file' parameters. By default these correspond to the table used to calculate COR2 for Suzaku, and a recalculation based on the IGRF (International Geomagnetic Reference Field) model (updated 2015) calculated for year 2016.0. The 'saafile' parameter determines the origin of the calculation of the satellite position with respect to the SAA, and contains the longitude and latitude of the vertices that define the SAA for each Hitomi instrument.
For the HXI ahmkehk also includes an additional screening definition stored in the column SAA2. The content of SAA2 is calculated using different definition of the SAA which longitute and latitude vertices are also in the 'saafile' CALDB file.

PARAMETERS
attfile [filename]
Input Hitomi attitude file.

orbfile [filename]

Input Hitomi orbit file, assumed to have the same format as Suzaku version 2 orbit files.

outfile [filename]
Output file in expanded housekeeping (EHK) format, containing a time series of quantities derived from the Hitomi instrument attitude and spacecraft orbit.

(tstart = 0.0) [real]
Start of mission time interval of the output file [s]. If either tstart or tstop is 0.0, the time interval is set equal to the time range of the reference file if there is one, or else to that of the attitude file.

(tstop = 0.0) [real]
End of mission time interval of the output file [s]. If either tstart or tstop is 0.0, the time interval is set equal to the time range of the reference file if there is one, or else to that of the attitude file.

(bintime = 1.0) [real]
Time sampling interval of the output file [s].

(textend = 0.0) [real]
Time added to the beginning and end of the output file time interval, preceding tstart and following tstop [s].

(reference = NONE) [filename]
Reference file supplying the actual time points to be calculated for the output file, overriding tstart and tstop. If NONE these are supplied by the input attitude file.

(timecol = TIME) [string]
Name of the column containing mission time in the reference file.

(attext = ATTITUDE) [string]
Name of the extension in the attitude file ('attfile') with pointing information.

(attform = EULER) [string: EULER|QUAT]
Format of the input attitude column in the 'attext' extension of 'attfile'.

(attcol = EULER) [string]
Name of the input attitude column in the 'attext' extension of 'attfile'.

(teldeffile = CALDB) [filename]
Name of the TELDEF file giving the coordinate systems of the SXI instrument, which is used fiducially to obtain sky coordinates. If set to CALDB (the default), the file is read from the calibration database.

(optaxis = 1215.5,1215.5,1215.5,1215.5,1215.5,1215.5,1215.5,1215.5) [string]
X and Y optical axis coordinates in FOC coordinates for HXI1, HXI2, SXI, and SXS instruments, respectively.

(cor2file = CALDB) [filename]
Name of the file that provides the cutoff rigidity as a function of terrestrial coordinates used to fill the COR2 column. If CALDB, the file used for Suzaku is read from the calibration database.

(cor3file = CALDB) [filename]
Name of the file that provides the cutoff rigidity as a function of terrestrial coordinates used to fill the COR3 column. If CALDB, the file calculated based on the International Geomagnetic Reference Field (IGRF) 2016 file is read from the calibration database.

(saafile = CALDB) [filename]
Name of the file that contains the vertices that define the SAA for each Hitomi instrument. If CALDB, the file is read from the calibration database.

(leapsecfile = CALDB) [filename]
Name of the file giving the table of leap seconds. If CALDB, the file is read from the calibration database.

(orbext = ORBIT) [string]
Name of the extension in the orbit file ('orbfile').

(orbcol = A) [string]

Name of the input orbit column in the 'orbext' extension of 'orbfile'.

(orbform = KEPLERIAN) [string]
Format of the input orbit velocity column in the 'orbext' extension of 'orbfile'.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

1. Create the extended HK file ah000506304.ehk from attitude and orbit files ah000506304.att.gz and ah000506304.orb.gz
ahmkehk attfile=ah000506304.att.gz orbfile=ah000506304.orb.gz outfile=ah000506304.ehk

2. Create the extended HK file ah000506304.ehk from attitude and orbit files ah000506304.att.gz and ah000506304.orb.gz using the quaternion column QPARAM in the attitude file.
ahmkehk attfile=ah000506304.att.gz orbfile=ah000506304.orb.gz outfile=ah000506304.ehk attcol=QPARAM attform=QUAT

NAME ahmkregion -- creates regions files showing the field of view of all Hitomi instruments

USAGE ahmkregion instrume ra dec roll

DESCRIPTION
'ahmkregion' creates special region files showing the field of view of each of the Hitomi instruments in each of the standard coordinate systems described in the Telescope Definition (TelDef) calibration files: ACTive, DETector, FOCal plane and SKY. The tool creates region files for a single instrument or one region file containing the field of view of all instruments. These files may be used for event filtering or in programs such as 'DS9' or 'ximage'. 'ahmkregion' may also create region files containing labels for the instrument and instrument segments and a script for reading the region files into 'ximage'

PARAMETERS
instrume [string]
Name of Hitomi instrument to be used. This must be one of SXS, SXI, HXI1, HXI2, SGD1, SGD2, or ALL. If set to one of the instrument, only the region file for that instrument is created. When 'instrume=ALL', the script creates individual instrument region files which are concatenated to make the unified region file and then deleted. To save separate instrument region files, use the 'outroot' parameter (see below).

ra [real]
The Right Ascension in decimal degrees of the nominal pointing. This must be in the range 0 <= ra <= 360 degrees.

dec [real]
The declination in decimal degrees of the nominal pointing, This must be in the range -90 <= dec <= +90 degrees.

roll [real]
The roll angle in decimal degrees between the SKY coordinate system and the FOC coordinate system. The roll angle is about the center of the SKY system and is measured counterclockwise from the positive SKY Y axis to the positive FOC Y axis.

(teldeffile = CALDB) [string]
Name of the Telescope Definition (TelDef) calibration data base (CALDB) file, which specifies the coordinate systems and transformation properties. If the parameter is set to CALDB, the default, the file is read from the calibration database. If a file name is provided, the TELESCOP keyword in the file must be 'HITOMI' and the INSTRUME keyword in the file must match exactly the values of the instrume parameters. Note that if 'instrume=ALL', then the only acceptable value for 'teldeffile' is CALDB, since multiple Hitomi TelDef files are read.

(inregion = NONE) [string]
By default, ‘ahmkregion' starts from a standard region file matched to the field of view of the instrument. 'inregion' allows for user-defined region files.

(outroot = NONE) [string]

Default names for the output region files are INST.SYS.box.reg, where INST is the instrument name and SYS the three-letter coordinate system name. 'outroot' allows for user-defined output region file names. This option is used when there is a need to preserve individual instrument region files when running with a script with 'instrume=ALL'.

(pixlist = 0-35) [string]
List of SXS pixels to be included in output region files. This is a comma-separated list for which the user can specify either single pixel numbers or a range of pixel numbers. This parameter is ignored if instrume is not SXS.

(outtextegion = yes) [boolean]
By default, 'ahmkregion' outputs two sets of region files: simple box-style regions with no text labels (INST.SYS.box.reg) and box-style regions with text labels (INST.SYS.text.reg). Setting 'outtextregion=no' suppresses the output of second set.

(outxco = yes) [boolean]
By default, 'ahmkregion' creates several instruction scripts (INST.SYS.xco) to read the region files into the program 'ximage'. Setting 'outxco=no' suppresses the production of these scripts.

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
1. Run ahmkregion for a single instrument with default settings
ahmkregion instrume=SXI ra=185.0 dec=-42.1 roll=35.0 teldeffile=CALDB

2. Run ahmkregion with instrume=ALL and suppress creation of text region files and xco scripts.
ahmkregion instrume=ALL ra=185.0 dec=-42.1 roll=35.0 outtextegion=no outxco=no

3. Run ahmkregion with instrume=SXS and a range of pixel values. Here, the user chooses pixel 0, pixel 2, and pixels 4, 5, 6 and 7.
ahmkregion instrume=SXS ra=25.0 dec=-42.1 roll=35.0 pixlist=0,2,4-7

NAME ahmktim -- Calculates the relation of Time Invariant vs Time and the GTI for when the GPS is on

USAGE ahmktim infile frqtemfile timfile outtimfile outgtifile

DESCRIPTION
'ahmktim' has two functions : a) calculate the Time Invariant vs. TIME relation (look-up table) and 2) write the GTI for when the GPS is on. The TI vs TIME look-up table is derived for each observation and it is used to assign time to all instruments and housekeeping. The relation is calculated either using the GPS , when this is on, or the on-board clock if the GPS is off (known also as Suzaku mode). To calculate TI vs TIME, 'ahmktim' uses the following files: a) the general HK file (see parameter 'infile'), b) the TIMing file (see parameter 'timfile') and c) the CALDB clock frequency vs temperature file (see parameter 'frqtemfile').

The relationship is calculated either using the GPS, when this is on, or the on-board clock if the GPS is off (known also as Suzaku mode). Both of these data are in the general HK file extensions (see parameter 'infile').

The GPS status is defined in the HK GPS extension (see parameter 'hkgpsext') by 4 flags written in columns (see parameters 'gpsacol', 'gpsbcol', 'gpsccol' and 'gpsdcol'). The combination of the 4 flags determines if the GPS is on or off. If the GPS is on the TI VS Time relation derived using the column S_TIME (rough time and correspond to the time when the packets are send to the telemetry) and L32TI (lower 32 bits of the Time Indicator) in the GPS extension. If the GPS is off, the relation is derived using the 'Suzaku mode', where the S_TIME in the GPS extension is matched with the closest time in the temperature extension (see parameter 'hktempext'). This temperature is then used to derive the frequency from the CALDB 'frqtemfile' and adjust the L32TI to when the GPS is off. The extension names for the GPS and Temperature as well as the column names depend on which on board computer , SMU (A or B), is disseminating the timing information.
The TI vs TIME relationship is added as second extension (EXTNAME=TIM_LOOKUP) of the TIMing input file ('timfile') which includes three columns: TIME, L32TI and STATUS. The STATUS column is a 10 bits column indicating the following status:
Bit 1:1 Legal or illegal 0=ok 1=not ok
Bit 2:3 Mode : GPS/suzaku transition 00=GPS mode 01=Suzaku mode 10=transition
Bit 4:5 Indicate status in transition region 00=monotonic 10=skip 11= duplicate
Bit 6:10 indicate error of illegal status

'ahmktim' calculates the GTI to when the GPS is on using the 4 flags written in HK GPS extension.

PARAMETERS
infile [filename]
Name of the general HK file containing the GPS and temperature extensions. This file is named ahgen_a0.hk1.

frqtemfile [filename]
Name of the calibration file containing the freq vs. temp relation. The parameter is set by default to CALDB.

timfile [filename]
Name of the input TIM file, containing the extension TIM_PACKETS.

outtimfile [filename]
Name of the output TIM file where the calculated TIM_LOOKUP extension is added.

outgtifile [filename]
Name of the output GTI file containing the intervals when the GPS is on.

(leapsecfile = REFDATA) [string]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

(hkgpsext = HK_GPS) [string]
Name of the extension with the GPS information in the general HK file (see 'infile'). This is HK_SMU_A_DHFS_SIB2GEN_dhfs_tlm_attseq and HK_SMU_B_DHFS_SIB2GEN_dhfs_tlm_attseq for the SMUA and SMUB respectively. The default is for SMUA.

(hktempext = HK_HCE) [string]
Name of the extension with the temperature information in the general HK file (see 'infile'). This is HK_SMU_A_AUX_HCE_HK2 and HK_SMU_B_AUX_HCE_HK3 for the SMUA and SMUB respectively. The default is for SMUA.

(timext = TIM_LOOKUP) [string]
Name of output extension in TIM file.

(gaptime = 2.0) [real]
Minimum value in sec to define a gap.

(stimecol = S_TIME) [string]
Name of the column containing S_TIME in the GPS extension.

(l32ticol = L32TI) [string]
Name of the column containing the L32TI in the GPS extension.

(tempcol = TEMP) [string]
Name of the column containing quartz temperature in the temperature extension. This is HCE_A_SENS_SMU_A_TEMP_CAL or HCE_B_SENS_SMU_B_TEMP_CAL for SMUA or SMUB respectively.

(packheadcol = PACKET_HEADER) [string]
Name of the column containing packet header. This counts subsequent packet and it is compared with their relative L32TI to determine whether or not there are dropped packets between two GPS measurements or large gap between two measurements.

(gpsacol = GPSA) [string]
Name of the column containing the GPS_A on/off flag. This is SMU_A_DHFS_TI_MNG_TIM_CRNT_TIM or SMU_B_DHFS_TI_MNG_TIM_CRNT_TIM for SMUA or SMUB respectively. The default is for SMUA.

(gpsbcol = GPSB) [string]
Name of the column containing the GPS_B on/off flag. This is SMU_A_DHFS_TI_MNG_TIM_GPS_SYC_STAT or SMU_B_DHFS_TI_MNG_TIM_GPS_SYC_STAT for SMUA or SMUB respectively. The default is for SMUA.

(gpsccol = GPSC) [string]
Name of the column containing the GPS_C on/off flag. This is SMU_A_DHFS_TI_MNG_TIM_AUT_SYC or SMU_B_DHFS_TI_MNG_TIM_AUT_SYC for SMUA or SMUB respectively. The default is for SMUA.

(gpsdcol = GPSD) [string]

Name of the column containing the GPS_D on/off flag. This is SMU_A_DHFS_TI_MNG_TIM_GPS_STAT or SMU_B_DHFS_TI_MNG_TIM_GPS_STAT for SMUA or SMUB respectively. The default is for SMUA.

(suzdrifttime = 100.) [string]
The maximum time intervals, in seconds, to calibrate the suzaku with the ground base measurements.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate a TIM_LOOKUP and a GPS GTI that are written in the file output.tim and gps.gti respectively using as the general HK file gen.hk , the CALDB file for the frequency vs temperature and the input tim file input.tim for the SMUA.

```
ahtime infile=gen.hk1 frqtemfile=CALDB timfile=input.tim outtim=output.tim \
outgtifile=gps.gti hkgpsext=HK_SMU_A_DHFS_SIB2GEN_dhfs_tlm_attseq \
hktempext=HK_SMU_A_HCE_HCE_A_SENS_STS tempcol=HCE_A_SENS_SMU_A_TEMP_CAL \
gpsacol=SMU_A_DHFS_TI_MNG_TIM_CRNT_TIM \
gpsbcol=SMU_A_DHFS_TI_MNG_TIM_GPS_SYC_STAT \
gpsccol=SMU_A_DHFS_TI_MNG_TIM_AUT_SYC  gpsdcol=SMU_A_DHFS_TI_MNG_TIM_GPS_STAT
```

NAME ahmodhkext - merges the extensions of a FITS file together into a single extension

USAGE ahmodhkext.pl infile outfile keyword

DESCRIPTION
This script merges several extensions of a FITS file into a single extension. The new extension is appended to the end of the FITS file. In addition to merging, rows in the new extension are sorted based on the value of the column specified in the parameter 'sortcol'.
Assumptions : Each extension in the FITS file must have identical column formats and must contain a keyword identical in all extensions.

The new extension contains an additional column. This column is named from the parameter 'colname'; if 'colname' is empty the column has the same name as the FITS keyword specified by the parameter 'keyword'. The column is populated with the value of this keyword from the extension from which the row was originally taken. This column is added to every source extension.

PARAMETERS
infile [file]
Input file containing the multiple extensions merged into one extension.

outfile [file]
Name of desired output file. If this is the same as infile, and if clobber is 'yes', the old file is rewritten.

keyword [string]
Keyword in each extension to be used to label data from that extension. This keyword must be present in all extensions to be merged. The keyword name is used to name the new column in the merged extension. The contents of that keyword populates the rows in the new column in the merged table. If the parameter 'colname' is filled, the new column is named with the content of ‘colname’.

(colname = "TESTKEY") [string]
Name of a new column in the merged extension. If empty, the column is named with the FITS keyword specified by the parameter 'keyword'.

(inputext = "*") [string]
Comma separated list of names of the extensions to process and merge. All extensions may be selected by using the asterisk ('*')

(outputext = "TARGETEXT") [string]
Name of the new extension containing the merged data.

(valuecol = "") [string]]

A list of values to populate the new column in the merged extension. There must be one entry for each respective source extension. If this parameter is filled, the new column is not populated with data from the 'keyword' parameter.

(tform) [boolean]]
Format of column to be created, default 'A'

(sortcol = S_TIME) [string]]
Name of column to sort resulting FITS file on.

(cleanup = yes) [boolean]]
Remove temporary files created in the process of merging the extensions. (yes/no)

(deletekey = yes) [boolean]]
Remove the keyword specified in the 'keyword' parameter from each source extension. (yes/no)

[ccldhm]

EXAMPLES
Merge two extensions of a file into one extension. Each extension contain a commond keyword TESTKEY. The value of the TESTKEY keyword in each extension is copied into a column named TESTKEY in the new merged extension. The merged extension is named TARGETEXT. The keyword TESTKEY is not deleted in the source extensions, although it is moved to the end of each header.
ahmodhkext.pl in.fits out.fits inputext='EXT2,EXT3' outputext=TARGETEXT keyword=TESTKEY deletekey=no

Merge two extension of a file into one. The merged extension is named TARGETEXT. The new column is named NEWCOLUMN. In NEWCOLUMN, every row that came from EXT1 have the value 'aa', and every row that came from EXT2 has the value 'bb'.
ahmodhkext.pl in.fits out.fits inputext='EXT2,EXT3' outputext=TARGETEXT colname='NEWCOLUMN' valuecol='aa,bb'

NAME ahpipeline - HXI/SGD/SXS/SXI reprocessing tool

USAGE ahpipeline indir outdir steminputs stemoutputs entry_stage instrument

DESCRIPTION
ahpipeline duplicates most of the pipeline (not trend data). It allows the user to run all or part of the pipeline processing and to vary the calibration files and filtering (screening) criteria used. A number of other pipeline processing parameters can also be changed.

Pipeline Stages

The pipeline is divided into 3 stages:

Calibration
Data screening
Product creation

Each stage may be run singly or in combination with the preceding stages. This is controlled by the entry_stage and exit_stage parameters.

HXI Stage 1 consists of the following ordered steps:
Run cams2att*
Run hxisgdsff
Run hxisgdpha
Run hxisgdexpand
Run hxievtid
Run coordevt
*Note: cams2att is automatically run twice. Once with parameters set for HXI1, the other with parameters set for HXI2.

SGD Stage 1 consists of the following steps:
Run hxisgdsff
Run hxisgdpha
Run hxisgdexpand
Run sgdevtid

SXI Stage 1 consists of the following steps:
For each event file:
(optional) Run coordevt on hot pixel file
Run coordevt
Run sxiphas
Run sxiflagpix
Run sxipi
Run sxipi
Create flickering pixel file
Filter SXI events STATUS[1]==b0
Run searchflickpix
Run coordevt on flickering pixel events
Run sxiflagpix with flickering pixels

SXS Stage 1 consists of the following steps:
(optional) Calculate GTIFOUNDALL (gtiinvert on GTILOST)
Run mxsgti
Calculate gaingti GTI file:
if linetocorrect eq 'MnKa':Merge GTIOBS GTITEL
else: Merge GTIOBS GTITEL GTIMXSFNON##
Run sxsanticopi
For each event file:
Run coordevt
Run sxsflagpix
Run sxssecid
Calculate GHF
Merge pointing and slew event files
Select merged file
if linetocorrect eq 'MnKa': PIXEL==12&&ITYPE<5&&gtifilter("<gaingti>")
else: PIXEL!=12&&ITYPE<5&&gtifilter("<gaingti>")
Run ftsort on filtered merged file
Run sxsgain on filtered, sorted merged file
For each event file:
Run sxspha2pi (GHF input)
Run sxsflagpix
Run sxssecid
Run sxsseccor
Run sxspha2pi (GHF input)
Run sxsperseus

The data screening (Stage 2) is identical to that in the production pipeline, when default parameters are used. ahpipeline uses the individual instrument pipelines to screen the data: hxipipeline, sgdpipeline, sxipipeline, sxspipeline. For details on the default screening applied to the events (respectively), see:
ahfilter - Create the EHK from attitude & orbital data and create the MKF from housekeeping data based on the CALDB mkfconf file
ahgtigen - Create the GTI from the EHK and MKF parameters based on CALDB selection file
ahscreen - Screen the data based on GTI and CALDB selection file

The product creation (Stage 3) is identical to that in the production pipeline, when default parameters are used. For HXI, SXI & SXS events extractor is run on SKY coordinates and a lightcurve, spectra and images are created for each cleaned event file. For SGD, extractor coordinates are set to NONE. No gif images are created.

Instruments

ahpipeline allows the processing of data on an instrument-by-instrument basis. This is controlled by the instrument parameter. Possible values are:
ALL - Process all data for all modules of all instruments
HXI - Process HXI data (both HXI1 and HXI2) only
HXI1 - Process HXI data HXI1 only
HXI2 - Process HXI data HXI2 only
SGD - Process SGD data (both SGD1 and SGD2) only
SGD1 - Process SGD data SGD1 only
SGD2 - Process SGD data SGD2 only

SXS - Process SXS data only
SXI - Process SXI data only

More than one value may be specified at one time.

Combination of HXI1 and HXI2 is allowed, is equivalent to HXI.

Combination of SGD1 and SGD2 is allowed, is equivalent to SGD.

INPUT

The input to ahpipeline is specified using (at minimum) the indir parameter. This should be specified as the top-level sequence directory, e.g.:

ahpipeline indir=/path/to/100039010 ...

Paths to specific housekeeping and satellite data files can be specified using the attitude, housekeeping, extended_housekeeping, makefilter, orbit and timfile parameters. But this should NOT be necessary if indir is specified correctly.

NOTE: Errors can occur if a subdirectory of a sequence directory is used for the indir parameter or if the output is stored within the sequence directory.

OUTPUT

Report
An output report is produced with the default name "ahpipeline.log". The name of the log can be changed with the logfile parameter. This lists all processing done in any particular run of ahpipeline. The level of detail of this report is controlled by the chatter parameter. (Default = "2", range 0 - 3.)

NOTE: Setting chatter to "3" or above can produce very large output reports.
Filenames, etc.

The number of output files depends on both pipeline processing stage(s) and the instrument(s) selected. All output files are written to the directory specified by the outdir parameter. The archive directory structure is NOT reproduced (i.e. all output files are in a single directory).

The names of files produced are the same as those found in the HEASARC archive. However the usual "ahXXXXXXXXX" prefix (where "XXXXXXXXX" is the sequence number) can be replaced by a character string set by the stemoutputs parameter. This defaults to the value set by the steminputs parameter.

PARAMETERS
indir [string]
Directory containing the input data. This should be the full or relative path to the top-level sequence directory, e.g.:
/path/to/805062010/
NOTE: Problems can arise if a sub-directory of the top-level sequence directory is used.

outdir [string]
Output directory used for all output data, as well as the report file. If clobber is not set, and this directory already exists, the task fails. This should NEVER be a sub-directory of the input directory (the top level sequence directory).
CAUTION: If clobber is set, and this directory already exists, then this task overwrites files as needed in this directory.

steminputs [string]
Stem for FITS input files, e.g. ah100039010. This string is used as the base filename for finding input files. For example, if steminputs=ah100039010, then to find the attitude file, ahpipeline matches the regular expression /ah100039010\.att(\..+)?$/ against all files found in the indir directory.When doing this matching, ahpipeline prefers files that match the original archive directory structure over files with the same name that do not. For example, if the files:
    /path/to/805062010/hxi/event_uf/ah805062010hxi_0_uf.evt
    /path/to/805062010/hxi/event_uf_2/ah805062010hxi_0_uf.evt
both exist, ahpipeline uses the first file, and not the second. If two files of the same name are found and neither one resides in an archive directory structure, ahpipeline exits with an error. NOTE: The above file finding issues can be avoided if the downloaded archive data set is left "pristine", by specifying an output directory OUTSIDE of the input directory tree.

(stemoutputs = DEFAULT) [string]
Base (stem) output name used for creating output files. If set to "DEFAULT", then steminputs is used.

entry_stage = 1 [1|2|3]
Entry stage, 1 or 2.
Stage 1: Re-calibrate unfiltered event files.
Stage 2: Start from existing unfiltered event files.

exit_stage = 2 [1|2|3]
Exit stage, 1 or 2.
Stage 1: Produces calibrated unfiltered event files.
Stage 2: Produces screened event files.

instrument = ALL [string]
Comma delimited list of which instruments to process. List can include the following values:
ALL - Process all data for all modules of all instruments
HXI - Process HXI data (both HXI1 and HXI2) only
HXI1 - Process HXI data HXI1 only
HXI2 - Process HXI data HXI2 only
SGD - Process SGD data (both SGD1 and SGD2) only
SGD1 - Process SGD data SGD1 only
SGD2 - Process SGD data SGD2 only
SXS - Process SXS data only
SXI - Process SXI data only

More than one value may be specified at one time:
Combination of HXI1 and HXI2 is allowed, is equivalent to HXI.
Combination of SGD1 and SGD2 is allowed, is equivalent to SGD.

verify_input [boolean]
Verify with ftverify (yes, no)

(attitude = DEFAULT) [string]
Attitude file

(housekeeping = DEFAULT) [string]
Housekeeping file

(extended_housekeeping = DEFAULT) [string]
Extended housekeeping file

(makefilter = DEFAULT) [string]
Makefilter file

(orbit = DEFAULT) [string]
Orbit file

(timfile = DEFAULT) [string]
Time file

(obsgti = DEFAULT) [string]
Observation GTI file

(calc_pointing = yes) [boolean]
Calculate nominal pointing ([yes]/no)

(calc_optaxis = yes) [boolean]
Calculate optical axis keywords ([yes]/no)

(create_ehkmkf = no) [boolean]
Create the EHK and MKF files (yes/[no])

(makeregion = no) [boolean]
Make a region file (yes/[no])

(ra = -999.99999) [real]
RA of nominal pointing [deg]

(dec = -999.99999) [real]
Dec of nominal pointing [deg]

(roll = 0.0) [real]
Roll of nominal pointing [deg]

(leapsecfile = REFDATA) [filename]
gen: Input leap second file (or CALDB, [REFDATA])

(selectfile = CALDB) [filename]
gen: Input file with the selection expressions

(mkfconf = CALDB) [filename]
gen: Input file with the selection expressions

(cor2file = CALDB) [filename]
Input cut-off rigidity Suzaku file (or CALDB)

(cor3file = CALDB) [filename]
Input cut-off rigidity IGRF 2016 file (or CALDB)

(saafile = CALDB) [filename]
CALDB

(hxi_start = 0.0) [real]
HXI CALDB start time

(hx1_teldef = CALDB) [filename]
coordevt: teldeffile hxi1

(hx2_teldef = CALDB) [filename]
coordevt: teldeffile hxi2

(hxi_remapfile = CALDB) [filename]
hxisgdsff/hxievtid: remapping file

(hxi_gainfile = CALDB) [filename]
hxisgdpha: PHA calibration functions

(hxi_badpixfile = CALDB) [filename]
hxisgdpha/hxievtid: readout channels

(hxi_fluorefile = CALDB) [filename]
hxievtid: Input fluorescence file

(hxi_enecutfile = CALDB) [filename]
hxievtid: Input energy cut file

(cm1_teldef = CALDB) [filename]
cams2att: CAMS1 teldeffile

(cm2_teldef = CALDB) [filename]
cams2att: CAMS2 teldeffile

(camstempxy = CALDB) [filename]
cams2att: CAMS temperature correction file

(sgd_start = 0.0) [real]
SGD CALDB start time

(sg1_teldef = CALDB) [filename]
coordevt: teldeffile

(sg2_teldef = CALDB) [filename]
coordevt: teldeffile

(sgd_remapfile = CALDB) [filename]
hxisgdsff/hxievtid: remapping file

(sgd_gainfile = CALDB) [filename]
hxisgdpha: PHA calibration functions

(sgd_badpixfile = CALDB) [filename]
hxisgdpha/hxievtid: readout channels

(sgd_fluorefile = CALDB) [filename]
sgdevtid: Input fluorescence file

(sgd_probseqfile = CALDB) [filename]
sgdevtid: sequence probability file

(sgd_probfovfile = CALDB) [filename]
sgdevtid: FOV probability file

(sxi_start = 0.0) [real]
SXI CALDB start time

(sxi_teldef = CALDB) [filename]
coordevt: teldeffile

(sxi_badpixfile = CALDB) [filename]
sxiflagpix: badpixfile

(sxi_maskfile = CALDB) [filename]
sxiflagpix: maskfile

(sxi_vtevnodd = CALDB) [filename]
sxipi: evenodd

(sxi_ctifile = CALDB) [filename]
sxipi: cti

(sxi_chtrailfile = CALDB) [filename]
sxipi: chtrail

(sxi_spthfile = CALDB) [filename]
sxipi: splitth

(sxi_gainfile = CALDB) [filename]
sxipi: gain

(sxi_patternfile = CALDB) [filename]
sxipi: grade

(sxs_start = 0.0) [real]
SXS CALDB start time

(sxs_teldef = CALDB) [filename]

coordevt: teldeffile

(sxs_coeftime = CALDB) [string]
sxssamcnt: Input file with arrival time coeffs

(sxs_pixdeffile = CALDB) [filename]
sxsflagpix: Input SXS electrical pixel map file

(sxs_gainfile = CALDB) [filename]
sxsupi: Input SXS gain coefficients file

(sxs_linefitfile = CALDB) [filename]
sxsdrift: Input calibration line file

(sxs_gainantfile = CALDB) [filename]
sxsanticopi: Input antico gain file

(sxs_delayfile = CALDB) [filename]
Input instrument delay file (or CALDB)

(hxi_mkflabel = HXISFFA1CAM) [string]
Label to use for HXI MKF GTI creation
For pseudo events "PSE" replace CAM in the label

(hxi_ehklabel = HXISFFA1CAM) [string]
Label to use for HXI EHK GTI creation
For pseudo events "PSE" replace CAM in the label

(hxi_evtlabel = HXISFFA1CAM) [string]
Label to use for HXI event screening
For pseudo events "PSE" replace CAM in the label

(sgd_mkflabel = SGDSFFA1#) [string]
Label to use for SGD MKF GTI creation. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(sgd_ehklabel = SGDSFFA1#) [string]
Label to use for SGD EHK GTI creation. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(sgd_evtlabel = SGDSFFA1#) [string]
Label to use for SGD event screening. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(sxi_mkflabel = SXI#SCI) [string]
Label to use for SXI MKF GTI creation. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxi_ehklabel = SXI#SCI) [string]
Label to use for SXI EHK GTI creation. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxi_evtlabel = SXI#SCI) [string]
Label to use for SXI event screening. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxs_mkflabel = PIXELALL1) [string]
Label to use for SXS MKF GTI creation

(sxs_ehklabel = PIXELALL1) [string]
Label to use for SXS EHK GTI creation

(sxs_evtlabel = PIXELALL1) [string]
Label to use for SXS event screening

(dattfile = datt.out) [string]
output datt file with drift corrections

(coordevt_startsys = LOWEST) [string]
Starting coordinate system

(stopsys = HIGHEST) [string]
Final coordinate system

(annaber = no) [string]
Apply annual aberration correction (yes, [no], INVERT)

(followsun = no) [boolean]
Recalculate the Sun position for each event (yes, [no])

(orbaber = no) [string]
Apply sat orbital aberration correction (yes, [no], INVERT)

(attinterp = LINEAR) [string]
Sky attitude interpolation method (LINEAR, CONSTANT)

(dattinterp = LINEAR) [string]
Delta attitude interpolation method (LINEAR, CONSTANT)

(attdt = 32.) [real]
Allowed margin for time extrapolation in attfile [s]

(dattdt = 0.5) [real]
Allowed margin for time extrapolation in dattfile [s]

(chkattgap = no) [boolean]
Limit attitude interpolation if gaps present (yes, [no])

(chkdattgap = yes) [boolean]
Limit delta attitude interpolation if gaps present ([yes], no)

(attext = ATTITUDE) [string]
Attitude extension

(attcol = QPARAM) [string]
Attitude column

(attform = QUAT) [string]
Attitude format ([QUAT], EULER)

(orbext = ORBIT) [string]
Orbit extension

(orbcol = VELOCITY) [string]
Orbital velocity column

(orbform = VECTOR) [string]
Orbital velocity format ([VECTOR], COMPONENTS, KEPLERIAN)

(coordevt_randomize = TELDEF) [string]
Randomize coordinates when rebinning ([TELDEF], yes, no)

(randsys = TELDEF) [string]
Starting system for randomization (or TELDEF)

(randscalesys = TELDEF) [string]
System to determine randomization amount (or TELDEF)

(infileext = EVENTS) [string]
Event extension

(inclfloatcol = no) [boolean]
Write non-rounded coordinate columns (yes, [no])

(inclfloatskycol = no) [boolean]
Write non-rounded sky coordinate columns (yes, [no])

(floatcolsuffix = _FLOAT) [string]
Suffix for non-rounded coordinate columns

(startwithfloat = no) [boolean]
Start with non-rounded startsys coordinates (yes, [no])

(blankcol = yes) [boolean]
Assign null values to columns not calculated ([yes], no)

(btnull = 255) [integer]
TNULL for byte (B) columns

(itnull = -999) [integer]
TNULL for short (I) columns

(jtnull = -999) [integer]
TNULL for long (J) columns

(ktnull = -999) [integer]
TNULL for long (K) columns

(sbtnull = 255) [integer]
TNULL for signed byte columns

(uitnull = -999) [integer]
TNULL for unsigned short columns

(ujtnull = -999) [integer]
TNULL for unsigned long columns

(outnsubcol = no) [boolean]
Output the PHA_NSUB column (yes/no)

(datamode = NONE) [string]
Substitute DATAMODE in place of event value (or NONE)

(outcalfile = NONE) [filename]
Output reconstruction tracing file (or NONE)

(startstep = 1) [integer]
Starting step of calculation (1-5)

(stopstep = 5) [integer]
Ending step of calculation (1-5)

(inext = EVENTS) [string]
Input extension

(outext = CAMS_OFFSETS) [string]
Output extension

(flipsign = no) [boolean]

Flip sign of output offsets and angles (yes/[no])

(prefiltfile1 = NONE) [string]
Prefiltered file for CAMS1

(prefiltfile2 = NONE) [string]
Prefiltered file for CAMS2

(filtoffset = NONE) [string]
Filtered offset file

(prefiltexpr = DSP_UP==1 && IS_SAMPLING==1) [string]
Expression to filter input files

(filtexpr = BAD_UNITS==0) [string]
Expression to filter offset file

(gtiexpr0 = BAD_UNITS==0) [string]
Expression to create GTI for both CAMS

(gtiexpr1 = BAD_UNITS==2) [string]
Expression to create GTI for CAMS1

(gtiexpr2 = BAD_UNITS==1) [string]
Expression to create GTI for CAMS2

(startsys = RAW) [string]
Starting coordinate system (cams2att)

(deltaxcol = DELTARAWX) [string]
Column with change in detector X coordinate

(deltaycol = DELTARAWY) [string]
Column with change in detector Y coordinate

(sincol = SINANGLE) [string]
Column with sine of rotation angle

(coscol = COSANGLE) [string]
Column with cosine of rotation angle

(outtracefile = NONE) [filename]
Output reconstruction tracing file (or NONE)

(numsignal = 48) [integer]
Maximum number of signals to analyze

(d10 = 3.2) [real]
Shortest distance between two adjacent pixels [mm]

(d1a1a = 5.0) [real]
Diag distance between two adjacent pixels in layer [mm]

(d1a1b = 5.0) [real]
Distance between two layers (CdTe-CdTe fluor) [mm]

(d1a2 = 14.0) [real]
Distance for combining Si-CdTe fluorescence [mm]

(d1a3 = 5.0) [real]
Distance for combining Si-Si electron scattering [mm]

(a = 3.0) [real]
Acceptance tolerance for F test in Step 2

(b = 3.0) [real]
Acceptance tolerance for G test in Step 2

(probaccept2 = 0.1) [real]
Probability threshold for M=2 acceptance in Step 3

(probaccept3 = 0.1) [real]
Probability threshold for M=3 acceptance in Step 3

(probaccept4 = 0.1) [real]
Probability threshold for M=4 acceptance in Step 3

(distz = 1000000.0) [real]
Very large distance of target object in Step 4 [mm]

(paraoffset0 = 1.6) [real]
Parameter used in calculating G[k,0]

(paraoffset1 = 1.0) [real]
Parameter used in calculating G[k,1]

(paraoffset2 = 1.0) [real]
Parameter used in calculating G[k,2]

(weight0 = 1.0) [real]
Parameter used in calculating G[k,0] for FOM

(weight1 = 0.0) [real]
Parameter used in calculating G[k,1] for FOM

(weight2 = 0.0) [real]
Parameter used in calculating G[k,2] for FOM

(weight3 = 0.0) [real]
Parameter used in calculating Prob[k] for FOM

(delgmethod = ANALYTIC) [string]
Method used to calculate Delta cos(theta)

(skipreco = no) [boolean]
Skip reconstruction of READALL/CALMODE events

(rejectbgo = no) [boolean]
Reject BGO events (yes/[no])

(occurrenceid = -1) [integer]
Occurrence to process (if >0)

(calc_hotpix = no) [boolean]
Run coordevt on hot pixel file (yes/[no])

(calc_modegti = yes) [boolean]
Calculate SXI data mode GTI ([yes]/no)

(colbound = -32768) [string]
TNULL, TLMIN, TLMAX for PHAS

(chipcol = CCD_ID) [string]
Chip column (or NONE)

(xcol = ACTX) [string]
X coordinate column

(ycol = ACTY) [string]
Y coordinate column

(chancol = PI) [string]
Pulse height column (or NONE)

(gradecol = GRADE) [string]
Event grade column (or NONE)

(grade = 0) [string]
Event grade for clean (or ALL)

(n_division = 1) [integer]
Divide total observation time into the given number

(cleanimg = no) [boolean]
Output cleaned image for debugging (yes, no)

(cellsize = 7) [integer]
Poisson clean cell size (odd integer > 1)

(impfac = 320) [real]
Factor for gamma function

(logprob1 = -5.6) [real]
Log Poisson probability threshold

(logprob2 = -5.6) [real]
Log Poisson probability threshold for second step

(iterate = yes) [boolean]
Iterate the second step Poisson clean (yes, no)

(flagedge = no) [boolean]
Zero chip edge pixels (yes, no)

(bthresh = 3) [integer]
Zero background threshold

(duration = no) [boolean]
Perform detailed search for flickering duration (yes, no)

(sigma = 3.0) [real]
Significance level for flickering duration

(firstchip = TLMIN) [string]
Min value for chip number

(lastchip = TLMAX) [string]
Max value for chip number

(xmin = TLMIN) [string]
Min value for X coordinate

(xmax = TLMAX) [string]
Max value for X coordinate

(ymin = TLMIN) [string]

Min value for Y coordinate

(ymax = TLMAX) [string]
Max value for Y coordinate

(chanmin = TLMIN) [string]
Min pulse-height value for clean (inclusive)

(chanmax = TLMAX) [string]
Max pulse-height value for clean (inclusive)

(outbadpix = no) [boolean]
Output bad pixel file (yes/[no]). This parameter is not a boolean in sxiflagpix but rather a filename. For ahpipeline and sxipipeline this is boolean to account for multiple files

(outbadimg = yes) [boolean]
Output bad pixel image ([yes]/no). This parameter is not a boolean in sxiflagpix but rather a filename. For ahpipeline and sxipipeline this is boolean to account for multiple files

(npixnbr = 1) [integer]
Pixel distance defining a neighbor

(nboundnbr = 1) [integer]
Pixel distance defining neighbor from CCD/window/segment boundary

(citrailnbr = 2) [integer]
Pixel distance trailing CI row

(ciprenbr = 1) [integer]
Pixel distance preceding CI row

(echoflag = yes) [integer
Flag CR echo pixels ([yes]/no)

(echomap = yes) [integer]
<dd>Output CR echo pixel fraction map (yes/[no])

(echonbr = 2) [integer]
Distance in pixels from a cosmic ray echo pixel to flag a neighbor pixel.

(echomin = 6) [integer]
Minimum number of events for the cosmic ray echo fraction calculation.

(echospth = 15) [integer]
Split threshold for cosmic ray echo fraction calculation.

(echofrac = 0.7) [float]
Minimum fraction of hits defining a cosmic ray echo pixel. For any pixel contained in at least 'echomin' events, if at least 'echofrac' of those events have a pulse height above 'echospth', then the pixel is considered a cosmic ray echo pixel.

(bad_status = 3:9,11,12,16:19,25:28,30) [string]
Bad status list, colons can used to specify a range (e.g. 1:3,5 = 1,2,3,5)

(copyphas = yes) [boolean]
Copy original PHAS before processing ([yes]/no)

(resetflags = yes) [boolean]
Reset all sxiflagpix STATUS flags ([yes]/no)

(hkext = HK_SXI_USR_USER_HK1) [string]
HK extension with video temperatures

(hkcolstem = SXI_USR_HKTBL_) [string]
Column name stem for video temperatures

(hkvideoid = A,B,B,B) [string]
Video card ID for gain correction of CCD1-4

(startcol = PHAS) [string]
Starting point of correction

(evnoddcor = yes) [boolean]
Enable even-odd correction [yes/no]

(chtrailcor = yes) [boolean]
Enable charge trail correction [yes/no]

(cticor = yes) [boolean]
Enable CTI correction [yes/no]

(gaincor = yes) [boolean]
Enable gain correction [yes/no]

(ctigrade = no) [boolean]
Use grade information in CTI correction [yes/no]

(copygrade = no) [boolean]
Copy existing GRADE and PHA columns [yes/no]

(phcut = CALDB) [string]
Pulse height cut for CTI correction, or CALDB

(badpixopt = 2) [integer]
Options for events with bad pixels: ignore bad pixels (1), null bad pixels (2), null whole event (3)

(spthiter = yes) [boolean]
Enable split threshold iteration [yes/no]

(spthcaldb = yes) [boolean]
Use split thresholds from spthfile [yes/no]

(spthoffset = 15.) [real]
Split threshold offset value (if spthcaldb = no)

(spthslope = 0.) [real]
Split threshold slope value (if spthcaldb = no)

(evtthre = DEFAULT) [string]
Event threshold (or DEFAULT)

(negthre = -5000) [integer]
Minimum PHAS value for normal event

(deltatime = 8) [integer]
Max allowed time gap in HK temp search [s]

(debugcol = no) [boolean]
Write out the debug columns [yes/no]

(sxs_resetflags = yes) [boolean]
Reset all sxsflagpix STATUS flags ([yes]/no)

(adrgti = REFDATA) [string]
Input ADR GTI file (or [REFDATA])

(acphaoffset = 0.0) [float]
Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+acphaoffset and +0.5+acphaoffset. So, when acphaoffset=0.5, the random offset is between 0 and 1.

(pxphaoffset = 0.0) [float]
Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+acphaoffset and +0.5+acphaoffset. So, when acphaoffset=0.5, the random offset is between 0 and 1.

(timecol = TIME) [string]
Time column

(stimecol = S_TIME) [string]
Name of S_TIME column

(tioncol = FWE_TI_LED#_ON) [string]
Input TI columns with LED on (#=1-4)

(tioffcol = FWE_TI_LED#_OFF) [string]
Input TI columns with LED off (#=1-4)

(plslencol = FWE_LED#_PLS_LEN) [string]
Input pulse length columns (#=1-4)

(plsspccol = FWE_LED#_PLS_SPC) [string]
Input pulse spacing columns (#=1-4)

(timeoncol = TIME_LED#_ON) [string]
Output LED-on time columns (#=1-4)

(timeoffcol = TIME_LED#_OFF) [string]
Output LED-off time columns (#=1-4)

(calctime = yes) [boolean]
Perform time assignment ([yes]/no)

(calcgti = yes) [boolean]
Produce GTI files ([yes]/no)

(afterglow = no) [boolean]
Add afterglow to fine GTI STOP times (no/[yes])

(dtdecay = CALDB) [string]
Afterglow time [s] (or CALDB)

(interp = twopoint) [string]
Interpolation method (NEAREST, TWOPOINT)

(margingti = yes) [boolean]
Create GTI between TSTART/TSTOP and first/last input GTI

(tstart = DEFAULT) [string]
Value to use for TSTART in seconds (or take from infile)

(tstop = DEFAULT) [string]
Value to use for TSTOP in seconds(or take from infile)

(dt = 0.) [real]
Time separation between input and output GTI (seconds)

(antpsp = A) [string]
Antico PSP to use for coincidence (A=PSPA B=PSPB)

(antshift = CALDB) [string]
Time shift [s] to apply to antico events (or CALDB)

(calcant = yes) [boolean]
Flag antico events ([yes]/no)

(antdtpre = CALDB) [string]
Delta time [s] preceding an antico event (or CALDB)

(antdtfol = CALDB) [string]
Delta time [s] following an antico event (or CALDB)

(antswitch = 1) [integer]
If =1 use antdtfol, =0 read delta-time from file

(antphathr = 71) [integer]
PHA threshold for antico events

(antdurthr = 2) [integer]
DURATION threshold for antico events

(calcctrec = yes) [boolean]
Flag recoil cross-talk ([yes]/no)

(ctrecdt = CALDB) [string]
Delta time [s] for flagging recoil cross-talk (or CALDB)

(calcprox = yes) [boolean]
Flag electrical cross talk ([yes]/no)

(proxdt = CALDB) [string]
Delta time [s] to define simultaneous events (or CALDB)

(calcctel = yes) [boolean]
Flag electrical cross talk ([yes]/no)

(cteldt = CALDB) [string]
Delta time [s] for flagging electrical cross-talk (or CALDB)

(ctelnear = 1) [integer]
Number of pixels for flagging electrical cross-talk

(calcctel2 = no) [boolean]
Flag electrical cross talk 2 ([yes]/no)

(cteldt2 = CALDB) [string]
Delta time [s] for flagging electrical cross-talk 2 (or CALDB)

(ctelnear2 = 1) [integer]
Number of pixels for flagging electrical cross-talk 2
(pxpithr = 600) [integer]
Events with PI values below this threshold are excluded from flagging checks given by the usepxpithr parameter.

(usepxpithr = ALL) [string]
A comma-delimited list specifying which flagging types should use the pxpithr parameter for excluding events. Allowed values in the list are ALL, NONE, PROX (proximity), CTEL (electrical cross talk), CTEL2 (2nd electrical cross talk), and CTREC (recoil cross talk). Events that do not belong to the types specified in the list are excluded from flagging regardless of their PI value.

(calcmxs = yes) [boolean]
Flag MXS pixels ([yes]/no)

(calc_gtilost = no) [boolean]
Calculate SXS lost off GTI (yes/[no])

(screenlost = no) [boolean]
Screen lost events (yes/[no])

(mxsdt = CALDB) [string]
Delta time [s] to extend MXS stop time (or CALDB)

(kalow = 5860.) [real]
Lower energy limit of Mn K-alpha for recoil PHA test [eV]

(kahigh = 5930.) [real]
Upper energy limit of Mn K-alpha for recoil PHA test [eV]

(kbeta = 6450.) [real]
Energy of Mn K-beta for recoil PHA test [eV]

(dtflag = no) [boolean]
Add delta-time columns for cross-talk and antico (yes/[no])

(dtprimary = CALDB) [string]
Time interval [ms] for primary (or CALDB)

(dtlowmid = CALDB) [string]
Upper time range [ms] for low secondary (or CALDB)

(dtmidhigh = CALDB) [string]
Upper time range [ms] for mid secondary (or CALDB)

(tol = 2.) [real]
Tolerance of time intervals [ns]

(regrade = no) [boolean]
Recalculate grade assignment (yes/[no])

(gaincoeff = H) [string]
Type of gain coefficients to use ([H]/M/L)

(linetocorrect = Mnka) [string]
Line to fit (HDU name in linefitfile)

(numevent = 250) [integer]
Maximum number of events in a single spectrum

(minevent = 150) [integer]
Minimum number of events in a single spectrum

(grpoverlap = 0.) [real]
Percentage of overlap between adjacent groups

(startenergy = -1.) [real]
Start energy [eV] of bin mesh (-1 = automatic)

(stopenergy = -1.) [real]
Stop energy [eV] of bin mesh (-1 = automatic)

(extraspread = 100.) [real]
Extend bin mesh energy range [eV]

(broadening = 1.0) [real]
FWHM Gaussian broadening of calibration line profile [eV]

(gridprofile = no) [boolean]
Calculate only the grid profile (yes/[no])

(fitwidth = yes) [boolean]
Fit spectrum width (yes/[no])

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE)

(spangti = no) [boolean]
Ignore GTI boundaries when binning spectra (yes/[no])

(usemp = no) [boolean]
Include Mp events when fitting (yes/[no])

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no])

(writeerrfunc = no) [boolean]
Output uncertainty functions (yes/[no])

(avgwinrad = 30) [real]
Radius of interval [binwidth] used to update average

(minwidth0 = 1.0) [real]
Smallest allowed initial value in width fitting [binwidth]

(maxitcycle = 5) [integer]
Maximum number of fitting iterations

(r2tol = .01) [real]
Convergence criterion for R^2

(searchstepshift = 2.) [real]
Step size when fitting shift [binwidth]

(maxdshift = 5.) [real]
Largest allowed deviation from initial guess of shift [binwidth]

(bisectolshift = .1) [real]
Tolerance of shift to stop bisection method [binwidth]

(searchstepwidth = 5.) [real]
Step size when fitting width [binwidth]

(maxdwidth = 10.) [real]
Largest allowed deviation from initial guess of width [binwidth]

(bisectolwidth = .2) [real]
Tolerance of width to stop bisection method [binwidth]

(minwidth = .5) [real]
Smallest width to allow in width fitting [binwidth]

(nerrshift = 100) [integer]
Number of shift values in uncertainty calculations

(nerrwidth = 100) [integer]
Number of width values in uncertainty calculations

(shifterrfac = 3.0) [real]

Factor for determining domain of shift uncertainty arrays

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays

(calcupi = yes) [boolean]
Calculate UPI column ([yes]/no)

(sxs_pulsefile = CALDB) [filename]
Input file with pulse amplitudes (or CALDB)

(phaout = PHA2) [filename]
Name of output PHA column

(sxs_scalefile = CALDB) [filename]
Input EPI scale file for cal-pix (or CALDB)

(secphacol = PHA) [string]
Input PHA column to use for secondary correction

(scaleepi = no) [boolean]
Scale EPI values using scalefile (yes/[no])

(scalegrade = 0) [string]
List of grades to apply scale factors

(calcpi = yes) [boolean]
Calculate PI column ([yes]/no)

(addepicol = EPI2) [string]
Output energy column with secondary correction

(method = FIT) [string]
Correction method (FIT or AVERAGE)

(extended = no) [boolean]
Use extended energy range (yes/[no])

(binwidth = 0.5) [real]
PI bin width for extended energy range [eV]

(offset = 0.5) [real]
Offset for first PI for extended energy range [eV]

(tlmax = 32767) [integer]
Maximum PI channel for extended energy range

(writetemp = no) [boolean]
Output temperature used for each event (yes/[no])

(dgfile = REFDATA) [file]
Input gain coefficients file

(offsetfile = REFDATA) [file]
calibration offset file

(outrange = NULL) [file]
How events are handled outside time range

(itypecol = ITYPE) [string]
ITYPE column

(ckctrec = no) [boolean]
Exclude events with recoil cross-talk (yes/[no])

(ckctel = no) [boolean]
Exclude events with electrical cross-talk (yes/[no])

(ckant = no) [boolean]
Exclude events with antico coincidence (yes/[no])

(ckrisetime = yes) [boolean]
Do not use events with RISE_TIME > 127 ([yes]/no)

(tempidx = 2) [integer]
Input temperature index for selecting gain

(ntemp = 3) [integer]
Number of temperatures from gain file to use in interpolation

(gapdt = -1.) [real]
Time [s] between events to define a gap (or

(extrap = no) [boolean]
Allow extrapolation when determining drift temperature (yes/[no])

(randomize = yes) [boolean]
Allow randomization ([yes]/no)

(seed = 0) [integer]
Random number generator seed (0=use system time)

(startdate =) [string]
Start date/time of ahpipeline (output)

(numerrs = 0) [string]
Number of errors from ahpipeline (output)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
The following command recalibrates (stage 1) and re-screen (stage 2) all HXI, SGD, SXS and SXI data for sequence 100039010 that currently resides in the directory /data/100039010/, and the output is stored in a directory called /data/100039010_reproc/:
ahpipeline indir=/data/100039010 outdir=/data/100039010_reproc entry_stage=1 exit_stage=2 steminputs=ah100039010 \
instrument=ALL

The following command re-screens (stage 2 only) HXI1 data for the same data set as in the previous example:
ahpipeline indir=/data/100039010 outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 steminputs=ah100039010 \
instrument=HXI1

NOTES
None, but see help for individual parameters above.

NAME ahscreen - Screen a science event file

USAGE ahscreen infile outfile expr

DESCRIPTION
'ahscreen' script screens science event files by applying an expression that filters on specified columns in the file, and/or by applying a GTI file (generally, the GTI created by the 'ahgtigen' task).

The input specified by the 'infile' parameter is an event file or list of event files (the name of the text file with the list, preceded by "@"). If multiple input files are specified, they are merged prior to screening. A single output file with filename given by the 'outfile' parameter is created. Any extensions that are not EVENTS are not copied to the final output file. The user may filter on a single GTI file or a list of GTI files by setting the 'gtifile' parameter (in the latter case by preceding the name of the list text file with "@").

If GTI is being filtered it must be specified in the gtifile parameter including any GTI attached to the input event file(s). In the case where a GTI file list is input and mergegti=NONE, the input GTI are nevertheless combined using AND. Otherwise the GTI from a list are combined according to whether 'mergegti' is AND or OR. If 'mergegti=AND' or 'mergegti=OR', the final GTI for the output file is attached and applied to the output file. If 'mergegti=NONE' and an output GTI file is created based on the outfile parameter but not attached to the output event file. If there is no input GTI file or GTI extension, no GTI screening is done.

Following the GTI screening, the filtering proceeds by applying a filtering expression on columns present in the event data file. Users may specify an expression via the 'expr' parameter or use a pre-existing CALDB filtering expression stored in the label file specified by the 'selectfile' parameter. The label file is a FITS file containing several expressions, each with a label that identifies the filtering details (the screening expression, and the type of file the expression is applied to) used to screen events and create the GTI. A user-input label file may also be used via the 'selectfile' parameter. A single label, or any combination of labels in the form of a comma-separated list may then be specified using the 'label' parameter.

Users may update the timing keywords (EXPOSURE, TSTART, TSTOP, etc.) using the 'upkeyword' parameter, and other keywords may be copied from the first input file onto the final file using the 'cpkeyword' parameter. The final output file is a single, filtered event file with fewer rows than the input file.

PARAMETERS
infile [filename]
Name of input event file or text file with list of files, preceded by "@" if multiple input files.

outfile [filename]
Name of output event file.

(gtifile) [filename]
Name of input gti file or text file with list of files preceded by "@" if multiple input files.

expr [string]
User-input expression applied to input file used to create GTI.

(selectfile = NONE) [filename]
Name of the select file or user input label file with labels and expressions.
If the parameter is set to CALDB, the file is read from the calibration database.
If the parameter is set to NONE, the task does not use this calibration information.

(label = NONE) [string]
Labels to read from label file specified by 'selectfile'. To input multiple labels, use a comma-separated list.

(mergegti = NONE) [string]
Attach GTI to event file and merge GTI mode (OR or AND) used in combining GTI files. If mergegti is set to NONE then an output GTI file is created and merging is done using an AND.

(cpkeyword = NONE) [string]
List of keywords to copy from input files, or ALL or NONE.

(upkeyword = yes) [boolean]
Update timing keywords from input file(s) in output file ([yes]/no).

(leapsecfile = REFDATA) [string]
Name of the leap second file. If the parameter is set to CALDB, the file is read from the calibration database; if REFDATA, the file is read from the REFDATA area.

(outexpr) [string]
Final expression used to create GTI. Output parameter only.

[ccldhm]

EXAMPLES
1. Run ahscreen with a simple screening expression.
ahscreen infile=sxs.evt outfile=sxs_cl.evt expr="ITYPE<5"
2. Run ahscreen with the event file CALDB expression labeled by PIXELALL.
ahscreen infile=sxs.evt outfile=sxs_cl.evt expr=NONE selectfile=CALDB label=PIXELALL
3. Run ahscreen with the event file CALDB expression labeled by PIXELALL, and an input GTI file to be merged with the input file GTI.
ahscreen infile=sxs.evt outfile=sxs_cl.evt gtifile=ahgengti.gti expr=NONE selectfile=CALDB label=PIXELALL mergegti=AND

NAME ahsxtarfgen - Creates an ancillary response file (ARF) for the SXS + SXT-S or SXI + SXT-I, accounting for the telescope effective area and detector efficiencies.

USAGE ahsxtarfgen telescope instrume emapfile qefile contamifile gatevalvefile rmffile onaxisffile onaxiscfile outfile regionfile skyregfile xrtevtfile

DESCRIPTION
'ahsxtarfgen' creates an ancillary response file (ARF) for the SXS + SXT-S or SXI + SXT-I. It takes as input an exposure map file ('ahexpmap'), a set of region files in detector coordinates, and an event file containing raytracing events for simulated photons injected into the telescope. The exposure map file is generated by the tool 'ahexpmap' and contains histograms based on discrete satellite attitude pointing. The region files are made by the script 'aharfgen' which uses the attitude information in the exposure map file and a region file in R.A./DEC coordinates to generate several region files in the detector coordinate system, one region file per attitude bin in the exposure map file. The raytracing event file is also made by the script 'aharfgen', which runs the raytracing code 'xrtraytrace' based on information in the attitude histograms in the exposure map file. The event file from previous runs of the raytracing may be used if the arf is recalculting using a different binning (the binning is defined by rmf), different region and whether or not the auxiliary transmission file is used.

The ARF includes the telescope effective area (for SXT-S in the case of the SXS, and SXT-I in the case of the SXI), quantum efficiency, extinction due to contaminants, filters, and in the case of the SXS, the gate valve if it is closed. The efficiencies are specified in CALDB files, and of these only the contaminants file has a time dependence. The SXI has just one filter transmission, and that is the contamination-blocking filter (CBF). The SXS has an optical blocking filter (OBF) but in addition also has a filter wheel mechanism that places a selectable filter in the X-ray telescope path at 921 mm above the focal plane. Aside from two open positions, the filter wheel may have one of four possible settings: Beryllium, Neutral Density, Polyimide, or an Fe 55 calibration source filter. If the user specifies CALDB for the filter wheel calibration file, 'ahsxtarfgen' takes the value of the keyword FILTER in the exposure map file and get the correct filter calibration file. In early observations of the mission, the SXS was operated with the gate valve closed. If the user specifies CALDB for the gate valve calibration file, 'ahsxtarfgen' takes the value of the keyword GATEVALV (which can have the value OPEN or CLOSED), and obtain the gate valve calibration file if the value of the keyword is equal to CLOSED.

PARAMETERS
telescop [string]
Mission name (value to write in header keyword TELESCOP for CALDB).

instrume [string]
Instrument for which the ARF is to be made: SXI or SXS.

emapfile [filename]
Name of the exposure map file (made by the tool 'ahexpmap'), that contains histograms of satellite attitude and related quantities, as well as "good time intervals" (GTI) for the attitude bins. In the case of the SXI or SXS there is also an effective exposure time image in the primary extension of the exposure map.

(qefile) [filename]
Name of the file containing the quantum efficiency (QE) for the detector. If the parameter is set to CALDB, the file is read from the calibration database. For the SXI the QE is combined with the optical blocking layer (OBL) transmission.

(obffile) [filename]
Name of the optical blocking filter file (needed for SXS only). If the parameter is set to CALDB, the file is read from the calibration database.

(fwfile) [filename]
Name of the filter wheel filter file (needed for SXS only). If the parameter is set to CALDB, the file is read from the calibration database.

contamifile [filename]
Name of the file containing information to calculate the transmission due to contaminants on the detector as a function of time, energy, and detector position. If the parameter is set to CALDB, the file is read from the calibration database.

(abund = "1.0") [string]
Relative abundances of contaminants. This number multiplies the abundances of all of the contaminant materials in the calibration file.

(cols = "0.0") [string]
Additional column densities for contaminants (1E18 cm^-2). The column densities of all of the contaminant materials in the calibration file are modified by adding the value of this parameter.

(covfac = "1.0") [string]
Partial covering factors for contaminant materials. The partial covering factors of all of the contaminant materials in the calibration file are multiplied by this number.

gatevalvefile [filename]
Name of the SXS gate valve calibration file. This file is only needed for SXS observations for which the value of the FILTER keyword is equal to CLOSED. The file accounts for the blocking and attenuation effects of the gate valve.

rmffile [file]
A response matrix file (RMF). For the SXS and SXI, it is only used to extract the energy grid that is used for the output ARF file.

onaxisffile [filename]
Name of the file and extension that contains the on-axis telescope effective area (appropriate for instrument), pre-calculated on a fine energy grid. If the parameter is set to CALDB, the file is read from the calibration database.

onaxiscfile [filename]
Name of file and extension that contains the on-axis telescope effective area (appropriate for instrument), pre-calculated on a coarse energy grid. If the parameter is set to CALDB, the file is read from the calibration database.

outfile [filename]
Name of the output ARF filename

regionfile [filename]
Names of standard SAO ascii region files, or the name of a file containing a list of names of such files. In the latter case, the format is "@aharfgen_regions.lis", where aharfgen_regions.lis is an ascii file in which each row contains the name of a region file. The coordinates of these region files are detector coordinates, and the region files specify the data extraction regions that correspond to the discrete satellite attitude intervals in the exposure map file ('emapfile') attitude histogram.

skyregfile [filename]
Name of the source region selection file, in R.A./DEC coordinates. The format is that of a standard SAO region file. This file must be the same one that was used in 'aharfgen'. Also, the image in the primary extension of the exposure map and 'skyregfile' must be self-consistent.

xrtevtfile [file]
Name of the raytracing event/history file that was created by 'aharfgen'.

(minphoton = 100) [integer]
The minimum number of photons that successfully reach the focal-plane, per raytracing energy grid point, that is acceptable to make a viable ARF. The number of focal-plane photons that contribute to the ARF must exceed minphoton for every energy, otherwise the program aborts.

auxtransfile [filename]
Name of the input auxiliary transmission file. This file is used to apply additional transmission that is not accounted in the telescope calibration files used by the raytracing. By default is set to NONE. To apply the auxiliary transmission users has either to input a file or set to CALDB.

(polydeg = "DEFAULT") [string]
Polydeg defines the polynomial order for the fitting. The allowed values are: DEFAULT and 1 to 5 for the HXI and 1 to 10 for the SXS and SXI. For the HXI the DEFAULT value of polydeg is set to 5. For the SXI and SXS, the DEFAULT value is set to the order tested internally for fitting stability.

[ccldhm]

EXAMPLES
1. Make an ARF file for a point source in the SXS given the R.A./DEC region selection file src_sky.reg, an exposure map file src_expmap.fits made by the tool 'ahexpmap', and a set of region files in detector coordinates that are derived from src_sky.reg using the exposure map src_expmap.fits and the tool 'aharfgen', which should be run prior to running 'ahsxtarfgen'. In the example, the gate valve is open so there is no need to specify the gate valve file. Each region file in detector coordinates corresponds to an attitude histogram bin in the exposure map file. In this example, the names of the region files in detector coordinates are listed, one file per row, in the ascii file aharfgen_region.lis. The raytracing event file that is required by the tool 'ahsxtarfgen' is made by the prior run of 'aharfgen'. Note that 'ahsxtarfgen' does not need any explicit information about the spatial distribution of the X-ray source because that is already factored into the raytracing event file. Since 'ahsxtarfgen' only uses the energy grid in 'rmffile' and not the actual matrix data, the particular response matrix (RMF) specified for 'rmffile' is not important. The ARF made by 'ahsxtarfgen' is calculated on the exact energy grid in 'rmffile'.

ahsxtarfgen telescop=HITOMI instrume=SXS emapfile=src_exmpmap.fits qefile=CALDB obffile=CALDB fwfile=CALDB\
contamifile=CALDB rmffile=sxs.rmf onaxiscfile=CALDB onaxisffile=CALDB outfile=src_1_arf.fits regionfile="@arfgen_region.lis" \
skyregfile=src_sky.reg xrtevtfile=src_1_raytrace.fits

The example run produces the output files: src_1_arf.fits

NAME ahsysinfo - Display system build information about an executable and/or about the system itself.

USAGE ahsysinfo.pl toolname

DESCRIPTION
Get information about current system (OS, Processor, Kernel, etc.) and which compiler was used to build tool.

PARAMETERS
None

EXAMPLES
1. Get information about the operating system itself
ahsysinfo.pl

2. Get information about the time assignment tool.
ahsysinfo.pl ahtime

NAME ahtime - Calculate times and populate the TIME column for all science and housekeeping data

USAGE ahtime infile outfile timfile lookupfile

DESCRIPTION
'ahtime' is a general mission task that assigns times to all science and HK data files. By default the calculated times are written in the column TIME (see parameter 'timecol') for science and HK data file and in the columns START and STOP for GTI files. Only for the HK files the times are also written as an UTC time in the columns YYYY, DDD, HH, MM, SS, US. The time calculation always requires the following supporting input files: a) the tim file (parameter 'timfile') containing the conversion TIME vs TI; b) the CALDB column definition file (parameter 'coldeffile') containing the columns names appropriate for the instrument; and c) the leap seconds file (parameter 'leapsec'). All science data also require the CALDB delay file (parameter 'delayfile') containing the delay added last to the calculated time. The SGD, HXI , SXS science data also require a look-up table stored in the instrument specific HK extension for fine time assignment. The delay file and the HK look-up table are not required when calculating the times for the HK file, but for the HXI and SGD. For the CAMS it is also necessary a CALDB file containing the offsets of the individual measurements relative to a time stamp. The framework to assign time is based on a three-steps process using two different look-up tables. For the SGD, HXI and SXS science, 'ahtime' first derives the fine time, in unit of L32TI, using a look-up table in the instrument HK (step 1). The fine time is assigned internally by the instrument processor. Afterwards 'ahtime' derives the time using the look-up table in the TIM files providing the TI vs TIME relation (step 2). Last it adds, when appropriate, a delay time between the on board computer and the instrument (step 3). For HK data, step 1 and 3 are skipped because there is not need of fine time assignment. The exceptions are the CAMS and MXS HK where a delay time is added to the times (step 3). For the SXI science the step 1 is skipped because the event data has already fine time information included, but calculates the times considering the DATAMODE (if WINDOW1 or WINDOW2 or WINDOW1BURST and/or WINDOW2BURST) and requires the several keywords to be present in the event file (TIMEDEL, TIMTRANB, EXPDEADB, TIMTRANA, EXPDEADA, FLUSHIMB, LASTDEAD, LASTDEL, NOMEXPO, TIMEPIXR). 'ahtime' updates the following header

keywords TSTART, TSTOP, DATE-OBS, DATE-END in the extension and in the primary that correspond to the smaller and largest values of the TIME column. If the file contains more than one extension as for the case of the HK, 'ahtime' calculates time in all extensions. The header keywords , in this case, are updated as follows: the primary header TSTART and STOP correspond to the lowest start and the largest stop considering all the extensions , while in each extension the TSTART and TSTOP are based on the smaller and largest time in the TIME column of the extension. The TIME column output from 'ahtime' is not time ordered.

PARAMETERS
infile [filename]
Name of the input file to calculate the times. This is either a science or HK file.

outfile [filename]
Name of output file with the times assigned.

timfile [filename]
Name of the input TIM file. This file contains the look-up table for the TI vs TIME conversion.

lookupfile [filename]
Name of the input HK file with local time information. This file contains the look-up table that is required only when times are calculated for the HXI, SGD, SXS science file and for the HXI/SGD HK.

(leapsecfile = REFDATA) [string]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

(coldeffile = CALDB) [string]
Name of the input file with the column names to use for each instrument and HK for the time calculation.
This is set by default to CALDB.

(delayfile = CALDB) [string]
Name of the input file with the time delays that are added to the science times.
This is set by default to CALDB.

(offsetfile = CALDB) [string]
Name of the input file containing offsets specific for the CAMS subsystem.
This is set by default to CALDB.

(timecol = TIME) [string]
Name of the column where the times are written. By default this is TIME.

(gticolums = START, STOP) [string]
Name of the column where the times are written in a GTI file. By default these are set to START and STOP.

(calctime = yes) [boolean]
Flag to perform time assignment. Valid only when processing an HK file.

(calcutc = yes) [boolean]
Flag to update UTC columns in the HK files. Valid only when processing an HK file.

(writekeys = yes) [boolean]
Flag to write the timing keywords in the header of the output file.

(interp = twopoint) [string]
Method of interpolation (nearest, twopoint, fourpoint)

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate times for an HK file, infile.hk using the tim file infile,tim. For HK the 'lookupfile' is set to NONE
ahtime infile.hk outfile.hk infile.tim NONE

2. Calculating times for a SXS science event file, infile.evt, using the tim file infile,tim. The SXS event requires a 'lookupfile' set to sxsinfile.hk
ahtime sxsinfile.evt outfile.hk infile.tim sxsinfile.hk

NAME ahtimeconv - Convert mission time in different formats

USAGE ahtimeconv intime, insys, inform, outsys, and outform

DESCRIPTION
'ahtimeconv' takes an input time (see parameter 'intime') in one time system (see parameter 'insys') and converts the time in another time systems (see parameter 'outsys'). The input and output time formats must be specified by the parameters 'inform' and 'outform'. The allowed formats are : calendar days as YYYY-MM-DDThh:mm:ss (c1), calendar day as YYYY:DDD:hh:mm:ss (c2), MJD (m), JD (j), seconds from epoch (s), and days from epoch (d).

The time systems supported are : Coordinated Universal Time (UTC), Terrestrial Time (TT), and Mission Elapsed Time (MET). The latter is defined as a number of seconds relative to an epoch given in the parameter 'epochtime'. The epoch time system and format may be given in the parameters 'epochsys' and 'epochform'.

The result of the time conversion is written in the parameter 'outtime'. For each converted time also the corresponding GPS time is calculated and written in the parameter 'gpstime'. 'ahtimeconv' requires the leap second FITS file (see parameter 'leapsec').

PARAMETERS
intime [string]
Input time in the time system and format specified in 'insys' and 'inform'.

insys [string]
Input system for the input time. Values allowed M for MET, T for TT and U for UTC.

inform [string]
Input format for the input time. Values allowed are : s seconds from epoch, d days from epoch, c1 calendar days as YYYY-MM-DDThh:mm:ss, c2 calendar days as YYYY:DDD:hh:mm:ss, j for Julian Day , m for Modified Julian Day.

outsys [string]
Output system for the output time. Values allowed M for MET, T for TT and U for UTC.

outform [string]
Output format for the output time. Values allowed are : s seconds from epoch, d days from epoch, c1 calendar days as YYYY-MM-DDThh:mm:ss, c2 calendar days as YYYY:DDD:hh:mm:ss, j for Julian Day , m for Modified Julian Day.

epochtime [string]
Epoch time in the time system and format specified by 'epochsys' and 'epochform'

epochsys [string]
Epoch system for the input epoch. Values allowed M for MET, T for TT and U for UTC.

epochform [string]
Epoch format for the input epoch. Values allowed are : c1 calendar days as YYYY-MM-DDThh:mm:ss, c2 calendar days as YYYY:DDD:hh:mm:ss, j for Julian Day , m for Modified Julian Day.

outtime [string]
Output time after the conversion.

gpstime [real]
GPS time [s] of the output time.

(leapsecfile = REFDATA) [string]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

numdigits [integer]
Number of decimal digits in sub-seconds

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Convert UTC time to mission time using default epoch:
ahtimeconv 2014-01-01T00:00:00 U c1 M s
ahtimeconv: input time: 2014-01-01T00:00:00 input time system: U input format: c1
ahtimeconv: output time: 63158400 output time system: T output format: s

2. Convert UTC time to mission time using custom epoch:
ahtimeconv 2014-01-01T00:00:00 U c1 M s epochtime=2010-01-01T00:00:00
ahtimeconv: input time: 2014-01-01T00:00:00 input time system: U input format: c1
ahtimeconv: output time: 126230400 output time system: T output format: s

3. Convert TT time to UTC
ahtimeconv 56658 TT m U c1
ahtimeconv: input time: 56658 input time system: T input format: m
ahtimeconv: output time: 2013-12-31T23:58:53.816000 output time system: U output format: c1

NAME ahtrendtemp - Calculate the quartz clock frequency vs temperature relation for the on-board clock

USAGE ahtrendtemp infile frqtemfile outtemp outfile

DESCRIPTION
'ahtrendtemp' calculates the quartz clock frequency vs. temperature relation using the in-flight data. The output of this task is used to monitor any deviation of the same relation derived from ground data and stored in CALDB. The input files are : a) the general HK file (see parameter 'infile', single or list of files are accepted), b) the CALDB clock frequency vs temperature file (see parameter 'frqtemfile'). The freq vs temperature is obtained by reading the frequencies corresponding to a temperature variation (see parameter 'tempresol') at a given time and averaging the frequencies in the delta temperature interval. The frequencies and the temperature are located in two separate extensions on the HK file and may be specified by the parameters 'quartzext' and 'tempext' respectively. The extension names for the GPS and Temperature as well as the column names depend on which on board computer , SMU (A or B), is disseminating the timing information. The timing information associated to the frequency and temperature are read for columns that may be specified in the parameters 'stimecol', 'l32ticol' and 'u32ticol'. The FITS output file (see parameter 'outfile') is created by cloning the format of the CALDB frequency vs temperature file and contains the following columns: TEMP (the clock temperature in Celsius), FREQ (the average frequency of the clock in Hz) and NPTTEMP (number of frequency points per temperature). 'ahtrendtemp' also outputs an ASCII temporary file (see parameter 'outtemp') where all frequencies (not averaged) and temperatures are written.

PARAMETERS
infile [filename]
Name of the general HK file that contains the quartz clock and temperature extensions. This file is named ahgen_a0.hk1. The task accepts either a string corresponding to a single FITS file or a string preceding by '@' corresponding to a text file containing a list of FITS file.

frqtemfile [filename]
Name of the calibration file containing the freq vs. temp relation. The parameter is set by default to CALDB.

outtemp [filename]
Name of the output file where all the frequencies , not averaged, and temperatures are written. This is an ASCII file.

outfile [filename]
Name of output file with the averaged frequencies and temperature are written. This is a FITS with the same format of the frqtempfile.

(starttime = ALL) [string]
Start time to start including data in the calculation. The time format should be entered as YYYY-MM-DDThh:mm:ss.

(stoptime = ALL) [string]
Stop time to stop including data in the calculation. The time format should be entered as YYYY-MM-DDThh:mm:ss.

(quartzext = HK_TIMNG) [string]

Name of extension in the general HK file (see 'infile') with the frequency information (quartz clock).
This is HK_SMU_A_DHFS_TI_MNG_block_get_ti_mng and HK_SMU_B_DHFS_TI_MNG_block_get_ti_mng for the SMUA and SMUB respectively. The default is for the SMUA.

(tempext = HK_HCE) [string]
Name of extension in the general HK file (see 'infile') with the temperature information.
This is HK_SMU_A_AUX_HCE_HK2 and HK_SMU_B_AUX_HCE_HK3 for the SMUA and SMUB respectively. The default is for the SMUA.

(quartzcol = RAW_QUARTZ_CLOCK) [string]
Name of the column with frequency information (quartz clock count) in the frequency extension.
This is SMU_A_DHFS_TI_MNG_TIM_TCAL_INF and SMU_B_DHFS_TI_MNG_TIM_TCAL_INF for the SMUA and SMUB respectively. The default is for the SMUA.

(l32ticol = L32TI) [string]
Name of the column with the L32TI times in the frequency extension.

(u32ticol = Quartz_U32TI) [string]
Name of the column with the U32TI times in the frequency extension.
This is SMU_A_DHFS_TI_MNG_TIM_TCAL_TIME and SMU_B_DHFS_TI_MNG_TIM_TCAL_TIME for the SMUA and SMUB respectively. The default is for the SMUA.

(tempcol = TEMP_CALC) [string]
Name of the column containing quartz temperature in the temperature extension.
This is HCE_A_SENS_SMU_A_TEMP_CAL or HCE_B_SENS_SMU_B_TEMP_CAL for SMUA or SMUB respectively.
The default is for the SMUA.

(stimecol = S_TIME) [string]
Name of the column containing the S_TIME times present in both frequency and temperature extensions.

(leapsecfile = REFDATA) [string]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

(tempresol = 1.0) [real]
Delta temperature, in unit of Celsius, over which the frequencies are averaged.

(stimemax = 500.) [real]
Maximum value of delta S_TIME where to search the temperature.

(averagemode = 1) [integer]
Mode to average frequencies. Support only 1) simple average

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate the freq vs temperature using the HK file gen.hk and the CALDB file for the frequency vs temperature. The outputs are written in the FITS file freqtemp.caldb.out and in the ASCII file dat.out.
ahtrendtemp infile=gen.hk frqtemfile=CALDB outfile=freqtemp.caldb.out outtemp=dat.out

---

NAME arftable - Create an effective area file and an ancillary response function (ARF) file from the output events from the raytracing code xrtraytrace, not including any detector efficiencies

USAGE arftable inxrtevtfile inrmffile outfileroot tgtOffaxis tgtAzimuth objecttype inimagefile objectheights objectradii imagerotangles xoffsets yoffsets regioncenter regionradius

DESCRIPTION

The 'arftable' script creates an effective area (EA) FITS file, and an ancillary response function (ARF) FITS file from input history event files that are created by the raytracing tool 'xrtraytrace'. 'arftable' allows simple region selection of events from the raytracing on the focal plane by specifying the center and radius of an event extraction circle. The script also allows one or more simple 2-D geometrical objects to be placed in the optical path between the bottom of the X-ray telescope and the focal plane. The objects may be in the form of planes with circular holes, or images. In this manner, objects such as the Hitmoi SXS gate valve, SXS filters, and HXI baffle may be modeled in a simple way. There is a possibility to write and event file to contain the events that are not stopped by the object between the telescope and focal plane (see parameter 'writeevtfile'). Note that if there are not objects between the telescope and the focal plane the event output file contains all the events of the input files.

The effective area in the EA and ARF files is due to the telescope only, and does not include any detector efficiencies. The only differences between the EA and ARF files are the energy grid and file format (see "OUTPUT" description below). The effective area values are identical in the two files. This script operates in two modes, depending on whether the value of the hidden parameter 'mergePtg' is set to 'yes' (default) or 'no'. If the value of 'mergePtg' is 'yes' (default) and there is only one input raytracing event file, the script merges events from all pointing directions in a single input raytracing event (or history) file (i.e. all values of off-axis and azimuthal angles relative to the telescope axial symmetry axis contribute to the effective area). If the value of 'mergePtg' is no the effective area is calculated by interpolating the results from raytracing runs performed at more than one pair of values of off-axis and azimuthal angle. This latter mode allows the effective area to be calculated quickly for X-ray source positions at different off-axis and azimuthal angles, without needing to run the 'xrtraytrace' tool repeatedly.

In the interpolation mode (if the value of 'mergePtg' is set to 'no', or if there is more than one input raytracing event file), the script creates an effective area for a targeted off-axis (theta) and azimuthal (roll) angle. These angles need not correspond to exact angles in any of the original raytracing runs since arftable interpolates the effective area for off-grid points. The script works by finding the off-axis and azimuthal angles immediately above and below the desired values, and creating an effective area for each. The script then calculates what the effective area should be for the desired angles. If the raytracing runs include an on-axis position (theta=0) the script can also calculate vignetting functions for every energy in the raytracing events (history) files (if the hidden parameter 'vignetting' is not equal to 'NONE'). However, note that if a vignetting file is created, the other files (EA and ARF) are not created.
INPUT

The input raytracing event file(s) must be in the same format as the output from the tool 'xrtraytrace', containing information about the event history of every raytraced photon and details of its path. The aperture, telescope description file (TDF), azimuthal angle grid, and energy grid must be consistent across each file. Each file may contain multiple off-axis angles, and may contain multiple roll angles. However, the same off-axis angle cannot be in multiple history files. Therefore the program is using at most two files: one for the theta below the target theta, and one for the theta above it. These two theta values could also be in the same file, and in this case the program would only be using one history file. In interpolation mode (if the value of 'mergePtg' is set to 'no'), the tool handles the cases where the target theta and target azimuthal angles may be on the grid of input angles, or between the angles. But they cannot be outside of the grid of angles. For example, if the input history files are created using off-axis angles 0, 10, and 20, then 0, 5, 10, and 18 (for example) would be valid target values of theta. Target theta values of 21 or 50 would not be valid. In merging mode (if the value of 'mergePtg' is set to 'yes'), the values of the input parameters 'tgtOffaxis' and 'tgtAzimuth' are ignored. However, note that if more than one input raytracing event file is given, the value of the parameter 'mergePtg' is ignored and the script runs in interpolation mode (equivalent to 'mergePtg' set to 'no').

An RMF (response matrix function) is required if the requested output is an ARF file. This because an ARF file must have the same energy grid as the input energy grid in the RMF file that the ARF file is going to be used with. The RMF filename is specified in the parameter 'inrmffile'. If set to NONE, the ARF file is not output.
OUTPUT

The output effective area is written to two FITS files: an EA file and an ARF file. The EA file has an identical format to the output EA file from 'xrtraytrace' (one column of energy, called ENERGY, and one column of effective area, called EFFAREA). The ARF file, readable by XSPEC, has the standard format of lower and upper energy bound columns (ENERG_LO and ENERG_HI respectively), and an effective area column called SPECRESP. When a vignetting file is desired, neither the EA file nor the ARF is created; only the vignetting file is created. The vignetting file contains functions of off-axis to on-axis effective area ratio versus energy. Last if the writeevtfile=yes than an event file is written out as describe above.

PARAMETERS

inxrtevtfile [filename]
Name of the input photon history (or event) file (in the format created by the raytracing tool 'xrtraytrace'). Alternatively, the name of an ASCII file many be entered, preceded by an '@' character, where the file contains the list of input files, one per line.

inrmffile [filename]

Name of the input RMF file (or NONE). If an RMF file is provided, the input energy grid from the RMF file is used to create the ARF file. The RMF file may also be a multi-extension line-spread function (LSF) file. Regardless of the type of RMF file, the energy grid is always obtained from the first extension. If 'inrmffile' is set to NONE, no ARF file is created (only the EA file is created).

outfileroot [string]
Root name for the output file EA and ARF files. If an ARF file is to be created, the file name is outfileroot.arf. The EA file name is outfileroot_ea.fits.

writeevtfile [boolean, (yes|no)]
Write or not an output event file. By default writeevtfile is set and the task does not write out any event file. Setting this paramater to yes the task writes an event file named using 'outfileroot' and adding '_evt.fits'. NOTE this is only useful if an object is placed bewteen the telescope and the focal plane to check which events are not stopped by the object and arrive to the focal plane.

(vignetting = NONE) [filename]
Name of the output vignetting file (or NONE). When a vignetting file is desired, the regular effective area files (EA and ARF) are not created; only the vignetting file is created. The file contains an extension with the ratio of the EA for each theta and azimuthal angle combination, to the EA at theta=0.

$$\frac{EA(E, theta, roll)}{EA(E, theta=0, roll)}$$

Because of the dependence on theta=0, one of the input history files must have raytracing results for theta=0.

(mergePtg = yes) [boolean, (yes|no)]
If 'mergePtg' is set to 'yes', raytracing results for all off-axis and azimuthal angles from a single input history event file are combined into a single effective area. If more than one raytracing event file is specified, then the value of the parameter 'mergePtg' is ignored and the action of the script is to produce an interpolated effective area for a pair of desired off-axis and azimuthal angles, equivalent to 'mergePtg' set to 'no'.

tgtOffaxis [real]
Target off-axis angle [arcmin]. This is the off-axis angle at which the desired effective area should be calculated in interpolation mode (i.e. if 'mergePtg' is set to 'no'). It must be within the range of the off-axis angles in the input file(s). For example, 'tgtOffaxis' cannot be smaller than the smallest off-axis angle in the provided input file(s). If 'mergePtg' is set to 'yes', then the value of 'tgtOffaxis' has no effect.

tgtAzimuth [real]
Target azimuthal (roll) angle [deg]. This is the azimuthal angle at which the desired effective area should be calculated in interpolation mode (i.e. if 'mergePtg' is set to 'no'). It must be within the range of the azimuthal angles in the input file(s). For example, 'tgtAzimuth' cannot be smaller than the smallest azimuthal angle in the provided input file(s). If 'mergePtg' is set to 'yes', then this parameter has no effect.

objecttype [string, (NONE|CIRCLE|IMAGE)]
Type of object to place in the optical path between the bottom (exit) of the telescope and the focal plane. Select NONE for no object in the path, CIRCLE for one or more circles, or IMAGE for one or more images. The heights of the circles above the focal plane and the radii of the circles are specified by the parameters 'objectheights' and 'objectradii' respectively (the 'objecttradii' parameter is ignored when 'objecttype' is not CIRCLE). The region outside a circle is treated as opaque, and the region on or inside a circle is treated as transparent. If 'objecttype' is IMAGE, the names of one or more FITS image files can be specified for the parameter 'inimagefile'. The images must be in the primary extension of each file and the X and Y units must mm. Image pixels that have a value greater than zero are treated as opaque and pixels that have values of zero are treated as transparent. The entire region outside the image boundaries is treated as opaque.

inimagefile [filename]
Name of an input image file modeling an object in the optical path between the telescope ad focal plane. Alternatively, the name of an ASCII file many be entered, preceded by an '@' character, where the file contains the list of input files, one per line.

objectheights [string]
A string containing one or more numbers, each number corresponding to the height (in mm) of a simple 2-D object above the focal plane. The heights of the objects must be listed in descending order.

objectradii [string]

A string containing one or more numbers, each number corresponding to the radius (in mm) of the "hole" in an otherwise opaque plane placed at a height above the focal plane that is specified by the corresponding number in the string parameter 'objectheights'. The parameter 'objectradii' is ignored if the parameter 'objecttype' is not CIRCLE.

imagerotangles [string]
A string containing one or more numbers, each number corresponding to the rotation angle (in degrees) of an image object in the optical path between the telescope and the focal plane. The rotation angle is zero if the X-axis of the image is aligned with the X-axis of the telescope and the rotation angle is positive if the image is rotated counter-clockwise in 'look-down' coordinates. The parameter 'imagerotangles' is ignored if 'objecttype' is not IMAGE.

xoffsets [string]
A string containing one or more numbers, each number corresponding to the offset (in mm) along the telescope X-axis of the center of an object in the optical path between the telescope and the focal plane. Note that the telescope axial symmetry axis has a value of 0.0 for the X-axis offset. Each number in 'xoffsets' is associated with a corresponding height above the focal plane in the string of numbers in the parameter 'objectheights'.

yoffsets [string]
A string containing one or more numbers, each number corresponding to the offset (in mm) along the telescope Y-axis of the center of an object in the optical path between the telescope and the focal plane. Note that the telescope axial symmetry axis has a value of 0.0 for the Y-axis offset. Each number in 'yoffsets' is associated with a corresponding height above the focal plane in the string of numbers in the parameter 'objectheights'.

regioncenter [string]
A string containing containing two numbers specifying the center (on the focal plane) of a circular selection region for raytracing events. The first number is the X-coordinate (in arcmin) and the second number is the Y-coordinate (in arcmin). A value of "0.0 0.0" for 'regioncenter' corresponds to the intersection of the telescope axial symmetry axis with the focal plane. Only events from the input raytracing files (specified by the parameter 'inxrtevtfile') that fall on or inside the circular region contribute to the output effective area. Note that if 'mergePtg' is set to 'no', or if there is more than one raytracing input event file, then the parameters 'regioncenter' and 'regionradius' are ignored.

regionradius [real]
The radius (in arcmin) of a circular selection region (on the focal plane) for raytracing events. Only events from the input raytracing files (specified by the parameter 'inxrtevtfile') that fall on or inside the circular region contribute to the output effective area. If no region selecton is desired (i.e. if all events are to be included), any negative value for 'regionradius' should be specified. Note that if 'mergePtg' is set to 'no', or if there is more than one raytracing input event file, then the parameters 'regioncenter' and 'regionradius' are ignored.

[ccldhm]

EXAMPLES

1) Create EA and ARF files from a raytracing photon history file named rt_evt.fits, with no objects in the optical path between the telescope and the focal plane, and no region selection, using an RMF file named src.rmf.

arftable.pl inxrtevtfile=rt_evt.fits inrmffile=src.rmf outfileroot=src mergePtg=yes objecttype=NONE regionradius=-1

The output files created is named src_ea.fits and src.arf.

2) Create EA and ARF files from an on-axis raytracing photon history file named rt_hx1_evt.fits, for a circular extraction region of radius 3 arcmin centered on the telescope symmetry axis, and including a simple model of the HXI1 baffle in the optical path between the telescope and the focal plane. The RMF file used is the 5-matrix line-spread function (LSF) ah_hx1_lsf_20140101v001.fits, which can be found in data/hitomi/hxi/bcf/ under CALDB. The HXI baffle is modeled as two planes at different heights above the focal plane, with the higher plane having a hole representing the baffle entrance, and the lower plane having a hole representing the baffle exit.

arftable.pl inxrtevtfile=rt_hx1_evt.fits inrmffile=ah_hx1_lsf_20140101v001.fits outfileroot=src mergePtg=yes objecttype=CIRCLE objectheights="622.0 124.0" objectradii="25.35 24.67" xoffsets="0.0 0.0" yoffsets="0.0 0.0" regioncenter="0.0 0.0" regionradius=3.0

The output files created is named src_ea.fits and src.arf.

3) Create an EA file by creating and interpolating effective area from multiple photon history files.

arftable.pl inxrtevtfile=@history_files.txt inrmffile=NONE outfileroot=eafile tgtOffaxis=35 tgtAzimuth=50 mergePtg=no

The input file assigned to 'inxrtevtfile' is an ascii file listing three history files, with off-axis angles 0 and 30 arcmin, 40 and 50 arcmin, and 60 and 70 arcmin respectively. Each history file uses the same azimuthal grid of 0, 45, and 90 deg, and same energy grid of 0.5, 2.0, 3.0 keV.

```
history_0_30.fits
history_40_50.fits
history_60_70.fits
```

Note that since the target off-axis is 35 arcmin, only the first two history files are used. If the target off-axis were 25, then only the first file would be used. The output EA file would be the effective area calculated at the target off-axis and azimuth, interpolated from the values in the input files.

4) Create an EA file using angles from a single photon history file.

```
arftable.pl inxrtevtfile=history_0_30.fits inrmffile=NONE outfileroot=eafile tgtOffaxis=0 tgtAzimuth=50 mergePtg=no
```

Here, the input file assigned to inxrtevtfile is a raytracing history file with off-axis angles 0 and 30 arcmin, azimuthal angles of 0, 45, and 90 deg, and energy values of 0.5, 2.0, 3.0 keV. The output EA file would be the effective area calculated at the target off-axis and azimuth. Since theta=0 is in the history file, only the azimuthal dependence would be interpolated.

NAME attconvert - converts attitude formats

USAGE
attconvert input inform

DESCRIPTION
'attconvert' is a mission-independent tool for converting attitude files from one format to another. It supports three formats: Quaternion [x, y, z, real], Z-Y-Z Euler angles and Pointing angles (Right Ascension, Declination, roll). Note that the tool renormalizes any input values for input values in quaternion format and rescales input angles as necessary (value outside range)

'attconvert' works in two modes:
1. Single point mode: 'input' parameter is a list of components separated by a comma. The 'inform' parameter is used to specify the format of the input list. The output formats are written both to stdout and to the output component parameters (the hidden parameters: 'quatx', 'quaty', 'quatz', 'quatr', 'eulerphi', 'eulertheta', 'eulerpsi', 'outra', 'outdec' and 'outroll' for the three different formats supported).

2. FITS file mode: In this mode, the 'input' parameter is a FITS file containing the column(s) to be converted. The column(s) name(s) for both the input and the output file are specified with the 'incol' and 'outcol' parameters respectively. If not present, the columns are created in the outfile.

PARAMETERS
input [string]
Either a comma-delimited list of the input attitude components or the attitude filename containing the columns to be converted.

inform = EULER [string]
Input attitude format. Allowed values are: QUAT, EULER or POINTING. The default is EULER.

(outfile='att_out.fits') [string]
Output file name. This parameter is needed and used only in FITS file mode.

(outform = EULER) [string]
Output attitude format(s) for FITS file mode. Allowed values are QUAT, EULER or POINTING or a list of these values. Use commas as separators in a list of more than one format. The order of the format names must match the order of the corresponding column names in the outcol parameter.

(outcol=QPARAM) [string]
Name of the output attitude column(s) used in FITS file mode. Use commas as separators in a list of more than one column. The order of the column names must match the order of the corresponding format names in the outform parameter.

(alignfile = "NONE") [string]

Name of the alignment calibration file. The alignment file specifies the rotation between the telescope and spacecraft axes and also specifies the roll convention. The roll may be extracted from the 3rd Euler angle "psi" by 'roll = ROLLSIGN * (psi - 90 + ROLL_OFF)'. If a file is not supplied and the parameter is either 'NONE' (Default) or 'STANDARD', the code uses an identity matrix as the rotation matrix and sets 'ROLL_OFF = 0'. If 'alignfile=NONE', 'ROLLSIGN = +1'. If 'alignfile=STANDARD', 'ROLLSIGN = -1'. This is the configuration for Hitomi, Swift, and Suzaku. If an alignment file is specified, the pointing angles correspond to the spacecraft z-axis pointing. If a TelDef file is specified as an alignment file, the pointing angles correspond to the instrument pointing. This type of file includes the roll definition and the alignment matrix components as standard keywords.

(attext = ATTITUDE) [string]
Name of the attitude extension of the attitude file, for FITS File Mode.

(incol = EULER) [string]
Name of the existing attitude column, for FITS File Mode.

(quatx = 0.0) [real]
Value of the output quaternion x-component, for Single Point Mode.

(quaty = 0.0) [real]
Value of the output quaternion y-component, for Single Point Mode.

(quatz = 0.0) [real]
Value of the output quaternion z-component, for Single Point Mode.

(quatr = 1.0) [real]
Value of the output quaternion real component, for Single Point Mode.

(eulerphi = 0.0) [real]
Value of the output first Euler angle (phi) in degrees, for Single Point Mode.

(eulertheta = 0.0) [real]
Value of the output second Euler angle (theta) in degrees, for Single Point Mode. When theta is 0 or 180 degrees, there are an infinite number of first and third Euler angle pairs that represent the same rotation. In these cases, the output Euler angles may be displayed in an alternative but equivalent representation.

(eulerpsi = 0.0) [real]
Value of the output third Euler angle (psi) in degrees, for Single Point Mode.

(outra = 0.0) [real]
Value of the output right ascension in degrees, for Single Point Mode.

(outdec = 90.0) [real]
Value of the output declination in degrees, for Single Point Mode.

(outroll= 270.0) [real]
Value of the output roll in degrees for Single Point Mode.

(clobber = no) [boolean]
Overwrites the existing output file if set to yes (yes/[no]).

[ccldhm]

EXAMPLES
1. Convert a single attitude quaternion to all formats using the default roll definition:
attconvert input="0,0,0.7071,0.7071" inform=QUAT
2. Convert a single attitude quaternion to all formats using the standard roll definition:
attconvert input="0,0,0.7071,0.7071" inform=QUAT alignfile=STANDARD
3. Convert the Euler angles in an attitude column to quaternions for a Suzaku attitude file:
attconvert input="ae107010010.att" outfile="ae107010010_eq.att" inform=EULER incol=EULER outcol=QUATERNION \
alignfile=STANDARD
4. Convert the quaternions in the attitude column QPARAM to both Euler and pointing angles for a Swift attitude file, using a Swift alignment matrix and roll definition:
attconvert input="sw00020215001sat.fits" inform=QUAT outfile="sw00020215001sat_pe.fits" outform=EULER,POINTING \

incol=QPARAM outcol=EPARAM,PPARAM alignfile="swalign20041115v012.fits"

NAME barycen - Apply barycenter corrections to X-ray timing data

USAGE barycen infile outfile orbfile ra dec

DESCRIPTION
'barycen' is a mission-independent tool to calculate barycenter corrections to X-ray timing data. The input FITS file must contain a column specified by the parameter 'timecol' (by default set to 'TIME'), assumed by 'barycen' to be the column to be corrected. Other time-related keywords in the initial FITS file header are also corrected. In all GTI extensions, any column name specified by the parameters 'startcol' ('TSTART' by default) and 'stopcol' ('TSTOP' by default) are also corrected. The corrected columns are overwritten in the output file. 'barycen' also needs the observatory orbit ephemeris file (defined in the 'orbfile' parameter), the target position (ra, dec), and the reference frame ('refFrame' parameter). If ra and dec are not provided or are out of range, the task extracts this information from the input FITS file header.

The task computes the time expressed by clocks at the Solar System Barycenter (SSB) (i.e., in TBD) and writes the corrected 'timecol' column in the output file specified by the parameter 'outfile'. This correction requires both geometric corrections (the projected light travel time to the Solar System Barycenter), and relativistic corrections such as Shapiro time delay and correction of clocks from the geocenter frame to the SSB frame.

The tool is suitable for any mission with an already corrected clock time; thus it should not be used for data from Swift or NuStar.

If the parameter 'debug' is set to 'yes', the following quantities are printed for each row of the input table: 'TIME', uncorrected time; 'BARYTIME', barycenter-corrected time; 'VEARTH', total barycentric Earth velocity; 'VEARTHX', 'VEARTHY', and 'VEARTHZ', X, Y and Z components of the barycentric Earth velocity; 'ERADVEL', projection of Earth velocity onto unit vector toward the observed target; 'TOTCORR', total barycenter time correction; the four terms that sum to 'TOTCORR': 'TTtoTDB', the difference between Terrestrial Time (TT) and Barycentric Dynamic Time (TDB), 'S/C_SSBC', light travel time from spacecraft to solar system barycenter, 'S/C_EARTH', correction due to Earth velocity, and 'GRAV', correction due to the gravitational field of the Sun. The (X, Y, Z) system is a right-handed Equatorial system where X points toward the Vernal Equinox and Z toward the North Celestial Pole. Velocity unit is km/s.

PARAMETERS
infile [filename]
Name of the input file.

outfile [string]
Name of the output file. The output file can be the same as the input file as long as the parameter 'clobber' is set to yes. In addition to the corrected columns and keywords listed under the infile description, the following keywords are added or modified: RA_TDB, DEC_TDB, TIMEREF and TIMESYS.

orbfile [filename]
Name of a FITS satellite orbit file. This file provides the satellite orbital position and velocity as a function of time used in calculating the barycenter correction. The orbit format in the orbfile must match the value of the orbform parameter.

ra [real]
Right Ascension in decimal degrees of the nominal pointing. This must be in the range 0 <= ra <= 360 degrees.

dec [real]
Declination in decimal degrees of the nominal pointing, This must be in the range -90 <= dec <= +90 degrees.

(orbext = PAR_ORBIT) [string]
Name of the FITS orbit file extension containing satellite orbit data.

(orbform = KEPLERIAN) [string]
Format in which the orbital velocity is provided in the orbit file. Three formats are supported. For the 'VECTOR' format, the position and velocity in the Earth-Centered Inertial (ECI) system are provided as two vector columns with three elements each. For the 'COMPONENT' format, the position and velocity in the ECI system are provided in three separate position columns and three separate velocity columns. For the 'KEPLERIAN' format, six separate columns contain the six Keplerian elements for the orbit solution.

(orbcol = A,E,I,AN,AP,MA) [string]

Names of the FITS columns containing orbit values in the orbext extension of the orbit file. If 'orbform=VECTOR', 'orbcol' is a string containing the names, in order, of the FITS columns containing the position values and the velocity values: e.g. 'orbcol=POSITION,VELOCITY'. If 'orbform=COMPONENTS', 'orbcol' must be a string containing six comma-separated column names, specifying in order the X, Y and Z components of the orbital position and then the VX, VY, VZ components of the orbital velocity, e.g. 'orbcol=X,Y,Z,VX,VY,VZ'. If 'orbform=KEPLERIAN', 'orbcol' must be a string containing six comma-separated column names, specifying in order the Keplerian orbital elements semi-major axis, eccentricity, inclination, longitude of the ascending node, argument of periapsis, and mean anomaly at epoch, e.g. 'orbcol=A,E,I,AN,AP,MA'.

(refframe = FILE) [string]
Coordinate reference frame. Allowed values are the string 'FILE', or the specific values 'FK5' (DE200) or 'ICRS' (DE405). If refframe='FILE' (default), then the tool reads the value of the 'RADECSYS' keyword in the input file.

(orbinterp = WEIGHTED) [string]
Method used to interpolate the position values in the orbit file. If 'orbinterp=WEIGHTED' (default), the tool calculates a weighted average of either the Keplerian elements ('orbform=KEPLERIAN') or the position values ('orbform=VECTOR' or 'orbform=COMPONENTS'). If 'orbinterp=TAYLOR', the tool uses a second-order Taylor expansion involving the position, velocity and time-derivative of the velocity to interpolate the orbital position. Note that the 'TAYLOR' interpolation method is not compatible with 'orbform=KEPLERIAN', and 'orbinterp' defaults to the 'WEIGHTED' interpolation method in such cases. If 'orbinterp=NEAREST', the tool does not interpolate the orbital position, but instead uses the orbital position value closest in time to the event time being corrected.

(timecol = TIME) [string]
Name of the FITS column in the event table containing time values.

(startcol = START) [string]
Name of the FITS column in the GTI table containing start time values.

(stopcol = STOP) [string]
Name of the FITS column in the GTI table containing stop time values.

(useweights = yes) [boolean]
Determines whether the code uses the values in a 'WEIGHTS' column of the orbit file when combining multiple orbit rows with the same time values. If 'useweights=yes', the tool derives a weighted average of the orbital positions and velocities for two rows with the same time. If 'useweights=no' or if the orbit file does not have a 'WEIGHTS' column, the tool uses only the values in the first of the two rows with the same time.

[ccldhm]

EXAMPLES
1. Run barycen using the default weighted orbit interpolation method and Keplerian orbital elements.
barycen infile=events.fits outfile=corrected_events.fits orbfile=orbitfile.fits ra=185.0 dec=-42.1

2. Run barycen using the Taylor orbit interpolation method and orbital positions and velocities supplied as vector columns.
barycen infile=events.fits outfile=corrected_events.fits orbfile=orbitfile.fits ra=83.63303 dec=22.01447 orbinterp="TAYLOR" \
orbext="ORBIT" orbform="VECTOR" orbcol="POSITION,VELOCITY" clobber= yes

3. Run barycen using the Taylor orbit interpolation method and orbital positions and velocities supplied as component columns.
barycen infile=events.fits outfile=corrected_events.fits orbfile=orbitfile.fits ra=83.63303 dec=22.01447 orbinterp="TAYLOR" \
orbext="ORBIT" orbform="COMPONENTS" orbcol="X,Y,Z,VX,VY,VZ" clobber= yes

---
NAME cams2att -- Computes a time-dependent delta-attitude file

USAGE cams2att infile1 infile2 outfile instrume

DESCRIPTION
'cams2att' calculates a time-dependent delta-attitude file using the CAMS telemetry data describing the wobbling of the Extended Optical Bench relative to the Fixed Optical Bench. The output may be used by 'coordevt' to calculate HXI ACT coordinates (wobble removed) given the RAW coordinates (subject to the wobble). 'cams2att' is instrument-specific and must be run separately on each of the two HXI units, HXI1 and HXI2.

'cams2att' runs in five steps to correct and filter the original CAMS data, calculate and filter intermediate offsets, and generate the attitude file. Any contiguous sequence of these steps can be run, using the 'startstep' and 'stopstep' parameters. The output file from Step

i is used as the input file to Step i+1. At each step, the intermediate file is saved only if the 'cleanup' parameter is set to 'no' and a name for that file is specified by the parameter corresponding to that step ('tempcorfile1/tempcorfile2', 'prefiltfile1/prefiltfile2', 'offsetfile', or 'filtoffset'). The clobber parameter applies to all intermediate files that are specified by name.

The five processing steps are:
Step 1. Correct the CAMS displacement data for temperature variations. This step reads the input 'X_RAW' and 'Y_RAW' columns and to the 'X' and 'Y' columns in the first intermediate output file.
Step 2. Filter the temperature-corrected files using ftselect and the expression specified by the 'prefiltexpr' parameter. This step removes rows with invalid CAMS displacement data.
Step 3. Run 'cams2det' for the HXI unit specified by the 'instrume' parameter. 'cams2det' calculates the offsets and rotations in HXI RAW coordinate space, based on the CAMS displacements.
Step 4. Filter the offset files using 'ftselect' and the expression given by the 'filtexpr' parameter. This step removes rows with poor quality calculations or those missing data from one of the CAMS units. It also generates a good-time interval file using maketime.
Step 5. Run 'det2att2' to convert the offsets and rotations in HXI coordinate space into quaternions.

PARAMETERS
infile1 [filename]
Name of input
FITS file containing the X and Y positions determined by the CAMS1 system.

infile2 [filename]
Name of input
FITS file containing the X and Y positions determined by the CAMS2 system.

outfile [string]
Name of output file containing the time-dependent delta-attitude quaternions for conversion from the HXI RAW to ACT system.

instrume [string: HXI1|HXI2|1|2]
Name of HXI unit for which output is calculated.

(cams1teldef = CALDB) [string]
Name of the CAMS1 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(cams2teldef = CALDB) [string]
Name of the CAMS2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(hxiteldef = CALDB) [string]
Name of the HXI1 or HXI2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(camstempxy = CALDB) [string]
Name of the CAMS temperature compensation file. If the parameter is set to CALDB, the file is read from the calibration database.

(startstep = 1) [integer]
The beginning step for processing within 'cams2att.' The five processing steps are listed in the description above. Possible values range from 1 to 5.

(stopstep = 5) [integer]
The final step for processing within 'cams2att.' The five processing steps are listed in the description above. Possible values are from 1 to 5.

(inext = CAMS_DATA) [string]
Name of the data extension in 'infile1' and 'infile2.'

(outext = CAMS_OFFSETS) [string]
Name of the data extension in the output file from step 1
(see parameters 'tempcorfile1' and 'tempcorfile2').

(flipsign = no) [boolean]
Flip the sign of the output offsets and angles. This parameter is used only for debugging by setting 'flipsign=yes'.

(tempcorfile1 = NONE) [string]

Name of the temperature-corrected CAMS1 displacement file. This may either be an output file if startstep=1 or an input file if startstep=2. The file is written by 'ftcalc' in Step 1, and it is read by 'ftselect' in Step 2. If startstep=1, 'tempcorfile1=NONE' and 'cleanup=yes', the file is deleted. If 'cleanup=no' the file is named 'tmp_tempcorfile1.fits'.

(tempcorfile2 = NONE) [string]
Name of the temperature-corrected CAMS2 displacement file.
See description of 'tempcorfile1.'

(prefiltfile1 = NONE) [string]
Name of the filtered, temperature-corrected CAMS1 displacement file. This may either be an output file if startstep<3 or an input file if startstep=3. This file is written by 'ftselect' in Step 2, and it is read by 'cams2det' in Step 3. If startstep=1 or startstep=2, 'prefiltfile1=NONE' and 'cleanup=yes', the file is deleted. If 'cleanup=no' the file is named 'tmp_prefiltfile1.fits'.

(prefiltfile2 = NONE) [string]
Name of the filtered, temperature-corrected CAMS2 displacement file.
See description of 'prefiltfile1.'

(offsetfile = NONE) [string]
Name of the intermediate unfiltered offset file. This may either be an output file if startstep<4 or an input file if startstep=4. This file is written by 'cams2det' in step 3 and read by the 'ftselect' in Step 4. If startstep=1, startstep=2, or startstep=3, 'offsetfile=NONE' and 'cleanup=yes', the file is deleted. If 'cleanup=no' the file is named 'tmp_offsetfile.fits'.

(filtoffset = NONE) [string]
Name of the intermediate offset file after filtering. This may either be an output file if startstep<5 or an input file if startstep=5. This file is written by the 'ftselect' in step 4 and read by 'det2att2' in Step 5. If startstep=1, startstep=2, startstep=3, or startstep=4, 'filtoffset=NONE' and 'cleanup=yes', the file is deleted. If 'cleanup=no' the file is named 'tmp_filtoffset.fits'.

(prefiltexpr = "DSP_UP==1 && IS_SAMPLING==1" [string]
Expression used by 'ftselect' to filter the temperature corrected CAMS data files 'tempcorfile1' and 'tempcorfile2.'

(filtexpr = "BAD_UNITS==0" [string]
Expression used by 'ftselect' to filter the 'offsetfile.'

(gtiexpr0 = "BAD_UNITS==0") [string]
Expression used to create GTI for both CAMS units.

(gtiexpr1 = "BAD_UNITS==2") [string]
Expression used to create GTI for CAMS1.

(gtiexpr2 = "BAD_UNITS==1") [string]
Expression used to create GTI for CAMS2.

(gtifile = camsboth.gti) [string]
Name of the CAMS GTI file output by Step 4. Extension i is created using 'gtiexpr(i-1)'.

(startsys = RAW) [string]
Origin coordinate system for transformation done by 'det2att2.' It can be either RAW or FOC.

(deltaxcol = DELTARAWX) [string]
Column name for change in X coordinate in the file 'filtoffset' read by 'det2att2' in Step 5.

(deltaycol = DELTARAWY) [string]
Column name for change in Y coordinate in the file 'filtoffset' read by 'det2att2' in Step 5.

(sincol = SINANGLE) [string]
Column name for sine of rotation angle in the file 'filtoffset' read by 'det2att2' in Step 5.

(coscol = COSANGLE) [string]
Column name for cosine of rotation angle in the file 'filtoffset' read by 'det2att2' in Step 5.

(cleanup = yes) [boolean]
Delete temporary files? ([yes]/no).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Run 'cams2att' with user-selected parameters for filtering and GTI expressions. Specify that the attitude is for HXI2. Also save the intermediate files produced by steps 2, 3, and 4 under user-specified names, but delete those from step 1:
cams2att infile1="CAMS1.fits" infile2="CAMS2.fits" outfile="HXI2_attitude.fits" \
cams1teldef="CALDB" cams2teldef="CALDB" hxiteldef="CALDB" camstempxy="CALDB" \
instrume="HXI2" clobber=yes prefiltexpr="FLAGS == 3 && ERROR == 0" \
filtexpr="((CALC_QUALITY==0 || CALC_QUALITY == 2) && (COSANGLE > 0.9))" \
gtiexpr0="BAD_UNITS==0 && CALC_QUALITY == 0" gtiexpr1="BAD_UNITS==2 && CALC_QUALITY == 0" \
gtiexpr2="BAD_UNITS==1 && CALC_QUALITY == 0" \
cleanup=yes tempcorfile1="NONE" tempcorfile2="NONE" prefiltfile1="CAMS1-filt.fits" \
prefiltfile2="CAMS2-filt.fits" offsetfile="nofilter_off.fits" filtoffset="postfilter_off.fits"

2. 'cams2att' can be started and stopped at different steps by employing on-default values of the 'startstep' and 'stopstep' parameters. Any combination of 'startstep' and 'stopstep' is allowed, as long as 'stopstep' is greater than or equal to 'startstep. 'The user must be careful with filenames when specifying non-default values for 'startstep 'and 'stopstep.' If the parameter 'startstep' is set to 1, 'infile1' and 'infile2' must be CAMS data but if the parameter 'startstep' is set to a number greater than 1, 'infile1' and 'infile2' are ignored, and appropriate intermediate file names must be specified. For example, if the parameter startstep is set to 4, the input file is specified using the 'offsetfile' parameter. In this example, the user corrects and filters the CAMS displacement files, but does not calculate the offsets or quaternions. To do this the user must specify startstep=1 and stopstep=2. Note that the 'outfile' parameter should be set to "NONE," since this file is not actually written. The 'prefiltfile1'and 'prefiltfile2' must be specified as existing files.

cams2att startstep=1 stopstep=2 infile1="CAMS1.fits" infile2="CAMS2.fits" outfile="NONE" instrum=HXI2 \
prefiltfile1="CAMS1-filt.fits" prefiltfile2="CAMS2-filt.fits" clobber=yes

3. In this example, the user calculates the quaternions from a filtered offset file. For this example, the user must specify startstep=5 and stopstep=5. \
cams2att startstep=5 stopstep=5 infile1="NONE" infile2="NONE" outfile="HXI_delta_att.fits" filtoffset="postfilter_off.fits" \
instrume=HXI1 clobber=yes

4. Run 'cams2att' with the default filtering and GTI expressions. Specify that the attitude is for HXI1. Start with Step 3 and end with Step 5. Also, delete all intermediate files except the one specified as input via the 'offsetfile' parameter. Note that in this example there is only one 'prefiltfile' specified, that for CAMS1. Therefore, the 'filtexpr' and 'gtiexpr0' must reflect that CAMS2 data is bad.
cams2att prefiltfile1="CAMS1-filt.fits" filtexpr="BAD_UNITS==2" gtiexpr0="BAD_UNITS==2" offsetfile="CAMS_offsets.fits" \
outfile="HXI1_attitude.fits" cams1teldef="CALDB" cams2teldef="CALDB" hxiteldef="CALDB" camstempxy="CALDB" \
instrume="HXI1" startstep=3 clobber=yes cleanup=yes infile1="NONE" infile2="NONE"

NAME cams2det - Calculate offsets and rotations in HXI coordinates using CAMS telemetry input

USAGE cams2det infile1 infile2 outfile instrume

DESCRIPTION
'cams2det' calculates HXI coordinate system offsets and rotations due to the wobbling of the Extended Optical Bench. The two input files are telemetry FITS files from each of CAMS1 and CAMS2. The output is a single FITS file with new columns containing the offsets in HXI RAW coordinates and the sine and cosine of the rotation angle. The output file can be converted into an attitude file by running the task 'det2att2' on the output file. 'cams2det' must be run separately on each of the two HXI units HXI1 and HXI2.

The routine matches the input TIME columns of infile1 and infile2. When a match is found, the routine uses data from both CAMS units to derive the offsets and rotation angle. If no match is found within a given tolerance (fixed in the software to 1 ms), the offset for that row is calculated based on a single CAMS unit and the rotation angle is set equal to zero.

The following columns are written to the output offset file:
TIME [s] is derived from the TIME column present in the two input CAMS data files.
DELTARAWX/DELTARAWY [pixels] is the component in the HXI RAW X/Y direction of the offset required to correct for the wobble of the HXI extended optical bench.
COSANGLE/SINANGLE is the cosine/sine of the rotation angle required to correct for the wobble of the HXI extended optical bench.

X1/X2 is the motion in the X-direction in the local coordinate system of the CAMS1/CAMS2unit. It is copied directly from the 'X' column in the input file infile1/infile2
Y1/Y2 is the motion in the Y-direction in the local coordinate system of the CAMS1/CAMS2 unit. It is copied directly from the 'Y' column in the input file infile1/infile2
JUMPX1/X2 is the difference between successive values of X1/X2.
JUMPY1/Y2 is the difference between successive values of Y1/Y2.
QUALITY1/QUALITY2 is copied directly from the QUALITY column of infile1/infile2.
XDISTANCE/YDISTANCE [mm] is the X/Y distance between the CAMS units
(X2 - X1)/(Y2 - Y1). This is calculated as an intermediate step in deriving the offset and rotation angle.
DELTASATX/DELTASATY [mm] is the offset equivalent to DELTARAWX/DELTASATY, but measured in the satellite coordinate system (SAT).
'BAD_UNITS' indicates which CAMS unit has valid data. If this equals to 0, both CAMS units have valid data, if it equals to 1, only CAMS2 has valid data and if equals 2, only CAMS1 has valid data.
CALC_QUALITY is generated internally to the code and is used to indicate one or more of the following situations.
CALC_QUALITY = 0: successful calculation of offsets and rotation (twist) angle.
CALC_QUALITY = 1: the rotation angle is not calculated because one CAMS unit has no good data.
CALC_QUALITY = 2: tiny twist angle fixed to cos(twist) = +-1, sin(twist) = 0 due to problem in the calculation of the twist angle.
CALC_QUALITY = 4: no calculation of offsets or twist angle possible.

PARAMETERS
infile1 [filename]
Name of the input CAMS1 telemetry FITS file containing the X and Y positions determined by the CAMS1 system.

infile2 [filename]
Name of the input CAMS2 telemetry FITS file containing the X and Y positions determined by the CAMS2 system.

outfile [filename]
Name of the output offsets file containing the offsets and rotation angle in HXI RAW coordinates.

(cams1teldef = CALDB) [string]
Name of the CAMS1 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(cams2teldef = CALDB) [string]
Name of the CAMS2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(hxiteldef = CALDB) [string]
Name of the HXI1 or HXI2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

instrume [string: HXI1|HXI2]
Name of the HXI unit for which the offsets are to be derived.

(inext = CAMS_DATA) [string]
Name of data extension in 'infile1' and 'infile2.'

(outext = CAMS_OFFSETS) [string]
Name of the extension of the output file to which the offset data is written.

(flipsign = no) [boolean]
Flip the sign of the output offsets and angles. This parameter is used only for debugging by setting 'flipsign=yes'.

(startsys = RAW) [string
Starting coordinate system. Valid values are RAW or FOC.

[ccldhm]

EXAMPLES
1. Calculate the offsets from the CAMS unit for motions of the HXI-1 instrument. The TelDef files for the CAMS are found using a CALDB query. The TelDef file for the HXI is specified on the command line.
cams2det infile1="CAMS1.fits" infile2="CAMS2.fits" outfile="HXI1_offsets.fits" cams1teldef="CALDB" cams2teldef="CALDB" \
hxiteldef="hxi1_teldef.fits" instrume="HXI1" clobber=yes

NAME camssim –

(1) Simulates motion of the Extensible Optical Bench (EOB) for testing CAMS-correction.
(2) Generates RAW coordinates associated with given HXI ACT coordinates and the simulated motion.

USAGE camssim infile1 infile2 outroothxi outrootcams

DESCRIPTION
'camssim' generates a set of sinusoidal motions of the Hitomi Extended Optical Bench (EOB), for both rotation and translation motions. From these simulated motions, 'camssim' calculates the amount by which each of the two CAMS units moves and, optionally the amount by which an HXI unit moves. 'camssim' has two different parts:
1. Generates simulated CAMS output files based on simulated EOB motion. The output files are compatible with all CAMS analysis tools and can be used to study in-flight motion. 'camssim' supports three types of statistical errors in the CAMS measurements but does not simulate CAMS malfunction, and always outputs two files with matched time columns and no missing data. Any simulation of missing data (filtering or modifying the file) should be done directly by the user.

2. Applies the simulated EOB motion to an HXI event file containing either real or simulated data. The simulated EOB motion calculated as described below is applied to the HXI ACTX and ACTY coordinates (no EOB motion) to generate RAWX and RAWY (with EOB motion). The motion is simulated as a combination of six sinusoids, two in each perpendicular linear direction (X and Y) and two in rotation (twist). The amplitudes of the six motions can be set independently, and two different frequencies can be set. The linear motions can also be chosen (using the 'typemotion' parameter) to be in phase, 90 degrees out of phase, or 180 degrees out of phase. The six independent amplitudes can be set as parameters 'xamp1,' 'xamp2,' 'yamp1', 'yamp2' (in mm, for translation) and 'ramp1,' 'ramp2' (in mrad, for twist). The two independent frequencies can also be set as parameters 'freq1' and 'freq2.' The output values (in the coordinates of the CAMS units) are 'delta_x', the resulting motion in the X-direction, 'delta_y', the motion in the Y-direction, and 'gamma', the twist angle. Using these definitions, the three possible types of motion are as follows, with time as the independent variable.
Linear Motion all in phase (typemotion=1).
delta_x = xamp1*cos(freq1*time) + xamp2*sin(freq2*time)
delta_y = yamp1*cos(freq1*time) + yamp2*sin(freq2*time)
gamma = ramp1*cos(freq1*time) + ramp2*sin(freq2*time)
Linear Motion X 180 degrees out of phase with Y and gamma (typemotion=2).
delta_x = -xamp1*cos(freq1*time) - xamp2*sin(freq2*time)
delta_y = yamp1*cos(freq1*time) + yamp2*sin(freq2*time)
gamma = ramp1*cos(freq1*time) + ramp2*sin(freq2*time)
Circular Motion. X and Y are 90 degrees out of phase (typemotion=3).
delta_x = xamp1*cos(freq1*time) + xamp2*cos(freq2*time)
delta_y = yamp1*sin(freq1*time) + yamp2*sin(freq2*time)
gamma = ramp1*cos(freq1*time) + ramp2*sin(freq2*time)
It is also possible to add random error to the simulated CAMS measurements. Calling "randn"
a normally-distributed random number and "freqerr" the frequency of the time-dependent part, and using the three input parameters 'randerr,' 'sinerr' and 'consterr', representing respectively the amplitudes of a random error, a time-varying error, and a constant error respectively. The errors are derived as:
error_x1 = randerr*randn + consterr*sin(freqerr*time) + sinerr
error_y1 = randerr*randn + sinerr*sin(freqerr*time) + consterr
error_x2 = randerr*randn + sinerr*sin(freqerr*time) + consterr
error_y2 = randerr*randn + sinerr*sin(freqerr*time) + consterr
The errors are applied to the output CAMS X/Y positions (delta_x, delta_y), but not to the rotation angle.

PARAMETERS
infile1 [filename]
Input HXI1 event file with ACTX and ACTY coordinate columns to be converted to RAWX and RAWY by applying the simulated motion. Set to "NONE" if this output is not required.

infile2 [filename]
Input HXI2 event file with ACTX and ACTY coordinate columns to be converted to RAWX and RAWY by applying the simulated motion. Set to "NONE" if this output is not required.

outroothxi [string]
Root of the name of the output HXI file(s). The output file name is outroothxi-n.fits, where n is either 1 or 2. If both 'infile1' and 'infile2' are given as valid files, then there are two output HXI files with suffixes n=1 and n=2, providing the output corresponding to

'infile1' and 'infile2,' respectively. If only one of 'infile1' or 'infile2' is given as a valid file, then there is only one output file, corresponding to whichever input file is given.

outrootcams [string]
Root of the name of then output CAMS files. The output file names are outrootcams-1.fits for CAMS1 and outrootcams-2.fits for CAMS2.

(hxiteldef1 = CALDB) [string]
Name of the HXI1 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(hxiteldef2 = CALDB) [string]
Name of the HXI2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(cams1teldef = CALDB) [string]
Name of the CAMS1 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(cams2teldef = CALDB) [string]
Name of the CAMS2 TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

(freq1 = 1.0) [real]
Frequency (Hz) of the 1st sinusoid in the simulated motion.

(freq2 = 1.0) [real]
Frequency (Hz) of the 2nd sinusoid in the simulated motion.

(xamp1 = 1.0) [real]
Amplitude (mm) of the X-motion (translation) for 1st sinusoid.

(xamp2 = 1.0) [real]
Amplitude (mm) of the X-motion (translation) for 2nd sinusoid.

(yamp1 = 1.0) [real]
Amplitude (mm) of the Y-motion (translation) for 1st sinusoid.

(yamp2 = 1.0) [real]
Amplitude (mm) of the Y-motion (translation) for 2nd sinusoid.

(ramp1 = 0.0) [real]
Amplitude (mrad) of the twist motion for 1st sinusoid.

(ramp2 = 0.0) [real]
Amplitude (mrad) of the twist motion for 2nd sinusoid.

(typemotion = 1) [int 1|2|3]
Type of motion 1 = X-Y in phase, 2 = X-Y out of phase, 3 = X-Y 90 degrees out of phase.

(deltatcams = 1.0) [real]
Time interval (s) in which the CAMS records five measurements.

(randerr = 0.240) [real]
Magnitude of CAMS error: random component (mm).

(consterr = 0.0) [real]
Magnitude of CAMS error: constant component (mm).

(sinerr = 0.0)[real]
Magnitude of CAMS error: sinusoidal component (mm).

(freqerr = 0.0) [real]
Frequency of the sinusoidal error component (Hz).

(seed = 0) [integer]

Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

(**[ccldhm]**

EXAMPLES
1. Generate HXI1 and HXI2 files from input event files and default parameters:
camssim infile1="hxi1_events.fits" infile2="hxi2_events.fits" outroothxi="outputhxi" outrootcams="outputcams" clobber=yes

2. Generate HXI1 file from input event files, no twist motion and user-supplied amplitudes and frequencies. The X and Y motions are to be 90 degrees out of phase.
camssim infile1="hxi1_events.fits" infile2="hxi2_events.fits" outroothxi="outputhxi" outrootcams="outputcams" freq1=10.0 \
freq2=0.5 xamp1=2.5 xamp2=3.6 yamp1=1.5 yamp2=2.1 ramp1=0.0 ramp2=0.0 typemotion=3 clobber=yes

3. Generate HXI1 file from input event files, same parameters as Example 2, but include user-supplied CAMS error parameters
camssim infile1="hxi1_events.fits" infile2="hxi2_events.fits" outroothxi="outputhxi" outrootcams="outputcams" freq1=10.0 freq2=0.5 \
xamp1=2.5 xamp2=3.6 yamp1=1.5 yamp2=2.1 ramp1=0.0 ramp2=0.0 typemotion=3 clobber=yes randerr=0.5 consterr=0.1 sinerr=0.95 \
freqerr=60.0

NAME coordevt - Convert events from one coordinate system to another.

USAGE coordevt infile[ext#] outfile teldeffile

DESCRIPTION
'coordevt' converts event coordinates in a FITS file from one system to another. The task requires two input files. The first is a file with an existing EVENTS extension containing a TIME and the coordinates column. The second file is a Telescope Definition (TelDef) file containing a COORDn keyword for each coordinate system present in the event file and a TRTYPEn keyword defining the transformation between two pairs of existing coordinate systems. The Teldef file also contains additional keywords describing each existing coordinate system (value of first pixel, scale, etc.).
'coordevt' is mission-independent and may currently process event files associated with the Swift, Suzaku and Hitomi missions. Please note that the default values for most of the hidden parameters defined below are mission-specific. They have been defined for Hitomi and need to be changed for the other missions that this task supports.

'coordevt' converts the lowest level coordinate present in the two input files to the highest level coordinates present in the teldef file or vice-versa (from the highest to the lowest). The transformations are carried out using double-precision floating point variables, but the calculated coordinates are converted to integers in the output file. If the hidden parameters ('inclfloatcol' and 'inclfloatskycol') are set to yes, the tool adds additional columns of unrounded event coordinates to the output file.
For any transformation involving SKY coordinate, an Attitude File, specified by the hidden parameter 'attfile', is required. The format of this file is specified by the hidden parameter 'attform'. 'coordevt' supports attitude files with the attitude provided as either quaternions or Euler angles but the file must contain a TIME column.
For any transformation to a coordinate system lower than SKY coordinates, each event coordinate is randomized within the pixel following the RANCOORD keyword in CALDB An additional auxiliary file is required in some cases, for example a transformation from the Hitomi HXI "RAW" to "ACT" coordinate systems. This file is specified by the hidden parameter 'dattfile'. The format of this 'dattfile" is fixed and must be in quaternion format.

PARAMETERS
infile [string]
Name of the input file containing the coordinates to be converted. The input file must have an existing EVENTS extension containing the coordinates to be transformed and a TIME column.

outfile [string]
Name of the output file. This file is a copy of the 'infile' with updated relevant keywords and values in the coordinate columns of the event extension. if the clobber parameter is set to 'yes' and the parameter 'outfile' is identical to that of 'infile', the input file is updated. If destination system columns do not exist in the input file, they are created in the output file. If columns corresponding to a destination coordinate system are present in the input file, then the values are overwritten in the output file. Any coordinate system in the chain before the 'startsys' is also copied, but systems after the 'stopsys' are either left intact or filled with a 'NULL' value. depending on the value of the parameter 'blankcol'. A NULL value is also written if the calculated value of a destination coordinate is outside the destination coordinate system range specified in the teldef file. All input file columns unrelated to coordinate transformations are unchanged and preserved in the outfile.

teldeffile [string]
Name of the Telescope Definition (TelDef), specifying the coordinate systems and transformation properties. If the parameter is set to CALDB, the default, a TelDef file for the telescope and instrument specified by the TELESCOP and INSTRUME keywords in the infile is read from the calibration database. The tool supports TelDef file format versions 0. through 0.2.

(attfile = NONE) [string]
Name of the attitude file used in the conversion to SKY coordinates. This keyword is required for any conversion involving SKY coordinates. The attitude may be provided as either quaternions or Z-Y-Z Euler angles but the attitude format in the attfile must match the value of the attform parameter. The special value 'attfile=IDENTITY' causes the identity attitude to be substituted for actual attitudes. This attitude is represented by the quaternion [0,0,0,1] or Euler angles [0,0,0] and is equivalent to the pointing ra=0.0, dec=+90.0, roll=+90.0. The 'ra' and 'dec' parameters should normally be set to 0.0, and +90.0 for the IDENTITY option. In addition, in order to make SKY coordinates equal to the next-lower coordinates (typically FOC), the 'roll' parameter must be set to +90. Other values of 'roll' rotate the SKY coordinates counterclockwise with respect to FOC by roll-90 degrees. For example, setting 'roll=0.0' rotates the SKY counterclockwise by -90 degrees (clockwise by 90 degrees) with respect to FOC, such that SKYX = -FOCY and SKYY = FOCX. The 'attform' parameter is ignored if 'attfile=IDENTITY'.

(dattfile = NONE) [string]
Name (or @list-of-names) of the auxiliary or "delta" attitude files needed for some coordinate transformations. The delta attitude file must be in quaternion format. A delta attitude file must be provided for each coordinate transformation of type 'SKYATT' in the teldef file, with the exception of the transformation to the SKY system. The special value 'dattfile=IDENTITY' causes the identity quaternion to be used for all delta attitudes.

(orbfile = NONE) [string]
Name of the FITS orbit file containing the satellite orbital velocity as a function of time. This file is only needed when the parameter 'orbaber' is set to yes and an orbital aberration correction is required . Two orbit formats (VECTOR and COMPONENT) are supported. The format VECTOR is applied to Swift while the format COMPONENT is valid for Suzaku. The orbit format in the 'orbfile' file must match the value of the orbform parameter.

(startsys = LOWEST) [string]
Name of the starting coordinate system of the requested conversion. When startsys is set to 'LOWEST', the default, the conversion begins with the lowest level system present, usually 'RAW', in the teldef file as identified by the parameter 'COORD0'.

(stopsys = HIGHEST) [string]
Name of the ending coordinate system of the conversion.
If stopsys is set to HIGHEST, then the coordinate transformation chain ends with the highest level system, usually 'SKY'.

(annaber = no) [string]
'annaber' allows to correct for annual aberration. The allowed setting are yes, no or invert. If set to no, the default, no correction is applied. If set to yes, the effects of annual aberration are taken into account when calculating the SKY coordinate values. If set to invert, multiplies the annual aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Annual aberration is the apparent bending of light due to the Earth's orbit around the Sun. This is at most a ~20.49 arcsec effect.

(followsun = no) [boolean]
If set to 'yes', 'followsun' uses the event time to compute the aberration for each event. This uses the MJDREF keyword (or pair of MJDREFI and MJFREFF keywords) along with the TIME column of the event extension to calculate absolute event times. If set to no (DEFAULT), the aberration is calculated using the time given by the 'MJDOBS' keyword for all events. Setting this to no is acceptable except for very long observations.

(orbaber = no) [string]
'orbaber' allows to correct for orbital aberration. The allowed setting are yes, no or invert. If set to no, the default, the orbital aberration is not corrected. If set to yes, the effects of orbital aberration are taken into account when calculating the SKY coordinates, provided an input orbit file is specified using the parameter 'orbfile'. If set to 'invert', multiplies the orbital aberration correction by -1 before applying it. The 'invert' option is only used for debugging. Orbital aberration is the apparent bending of light due to the satellite's orbit around the Earth. For a satellite in low-earth orbit this is at most a ~5 arcsec effect.

(attinterp = LINEAR) [string]
Spacecraft attitude interpolation method: LINEAR or CONSTANT. When this parameter is set to LINEAR (the default), the attitude is linearly interpolated to the event time. When it is set to CONSTANT, the attitude is taken from the nearest attitude record.

(dattinterp = LINEAR) [string]

Delta attitude interpolation method: LINEAR or CONSTANT. When this parameter is set to LINEAR (the default), the delta attitude is linearly interpolated to the event time. When it is set to CONSTANT, the delta attitude is taken from the nearest attitude record.

(attdt) [real: 32.0]
Maximum time interval (in seconds) allowed to extrapolate the spacecraft attitude before the first or after the last row in the attitude file. Events beyond this time margin have their coordinates set to a null value.

(dattdt) [real: 0.5]
Maximum time interval (in seconds) allowed to extrapolate the auxiliary attitude (specified by the keyword 'dattfile' before the first or after the last row in the attitude file. Events beyond this time margin have their coordinates set to a null value.

(chkattgap = no) [boolean]
Used only for interpolation. If set to no (the default), always interpolate the attitude at the event time. If set to yes, event times that fall within the sky attitude time range but not within the 'attdt' of the nearest attitude time is be set to null.

(chkdattgap = no) [boolean]
Used only for interpolation. If set to no (the default) the task always interpolates the attitude at the event time. If set to yes, event times that fall within the delta attitude time range but not within the 'dattdt' of the nearest attitude time is be set to null.

(attext = ATTITUDE) [string]
Name of the FITS attitude file extension containing satellite attitude data.

(attcol = QPARAM) [string]
Name of the FITS column containing attitude values as a vector in the attext extension of the attitude table.

(attform = QUAT) [string)
Format of the attitude table. Two formats are supported. The QUAT format is a quaternion [x, y, z, real]. The EULER format is a Z-Y-Z Euler angle trio [phi, theta, psi] in degrees.

(orbext = ORBIT) [string]
Name of the FITS orbit file extension containing satellite orbit data.

(orbcol = VELOCITY) [string]
Name(s) of the FITS column(s) containing orbit values in the orbext extension of the orbit file. This parameter is linked to the parameter 'orbform'. The only acceptable values for 'orbcol' are: VELOCITY, VECTOR or COMPONENTS. The default is 'VELOCITY'. If 'orbform=VECTOR', then 'orbcol' is the name of the single FITS column containing the orbit values as a vector. If 'orbform=COMPONENTS', then 'orbcol' must be a string containing three comma-separated column names, specifying in order the X, Y and Z components of the orbital velocity. These columns must be scalar. Velocity must be in km/s in an Earth-Centered Inertial system.

(orbform = VECTOR) [string]
The format in which the orbital velocity is provided in the orbit file. Three formats are supported. For the VECTOR format (DEFAULT), the velocity is provided as a vector column with three elements (X, Y and Z in Earth-Centered Inertial (ECI) system). For the COMPONENT format, the velocity is provided in three separate columns. For the KEPLERIAN format, the velocity is derived from the six provided Keplerian element columns.

(randomize = TELDEF) [string]
If this parameter is set to 'no', 'coordevt' assumes that each event occurred at the center of its coordinate pixel. If 'randomize=TELDEF', the default, randomization depends on the keyword RANCOORD in the TELDEF file ('RANCOORD = NONE' disables randomization). If 'randomize=yes', the coordinates from system 'randsys' (see below) onward is calculated assuming a random location within the randsys system pixel. This parameter only controls randomization in transformations previous to the transformation to the SKY system.

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(randsys = TELDEF) [string]
Name of the starting coordinate system for which randomization is performed, as long as randomization is enabled by the 'randomize' parameter. If 'randsys=TELDEF', the value of the RANCOORD keyword (if present)in the TELDEF file specifies this system.

(randscalesys = TELDEF) [string]
Name of the coordinate system whose pixels determine the size of the randomization if randomization is enabled by the 'randomize' parameter. If 'randscalesys=TELDEF', the value of the 'RAN_SCAL' keyword, if it exists, in the TELDEF file is used. If the

'RAN_SCAL' keyword does not exist in the TELDEF file, the value of the 'randsys' parameter is used. 'randsys' and 'randscalsys' may not be the same coordinate system.

(infileext = EVENTS) [string]
Name of the infile extension containing the event table.

(timecol = TIME) [string]
Name of the FITS column in the event table containing time values.

(inclfloatcol = no) [boolean]
If this parameter is set to 'yes', an additional column of unrounded coordinate values is included in the output event file for each coordinate of each system after the startsys system. Enabling this parameter is useful for stringing multiple coordevt runs together (e.g., convert RAW to DET in one run and DET to SKY in a second run) without loss of precision due to the normal rounding of coordinates. Use 'startwithfloat' set to yes for runs after the first so that the unrounded coordinates from the previous coordevt run are used as the starting coordinate values for a subsequent run. Rounded coordinate columns are written regardless of the value of this parameter. Unrounded SKY coordinate columns are included if either of 'inclfloatcol' or 'inclfloatskycol' is set to 'yes'.

(inclfloatskycol = no) [boolean]
If this parameter is set to 'yes', an additional column of unrounded coordinate values is included in the output event file for each SKY coordinate. Rounded SKY coordinate columns are included regardless of the value of this parameter. Unrounded SKY coordinate columns are included if either of 'inclfloatcol' or 'inclfloatskycol' is set to 'yes'.

(floatcolsuffix = _FLOAT) [string]
This parameter sets the suffix used in all unrounded coordinate column names. If the rounded coordinate column name is 'DETX' and 'floatcolsuffix=_FLOAT', then the corresponding unrounded coordinate column name is 'DETX_FLOAT'. This parameter is used only when either of 'inclfloatcol' or 'inclfloatskycol' is set to 'yes'.

(startwithfloat = no) [boolean]
This parameter is set to 'yes' to use the unrounded coordinate column values (e.g., floating point values from 'DETX_FLOAT' and 'DETY_FLOAT' columns) as the starting coordinate columns. The 'inclfloatcol' parameter must be set to 'yes' for 'startwithfloat' to have an effect. When either of 'inclfloatcol' or 'startwithfloat' is disabled, rounded coordinate values (e.g., integer values from DETX and DETY columns) are used as the starting coordinate values. Enabling this parameter is used for debugging.

(blankcol = yes) [boolean]
This parameter is set to 'yes' to fill the coordinate columns after the 'stopsys' coordinate system with NULL values. If blankcol is set to no, such columns is not changed. The behavior is somewhat different depending on what coordinate columns already exist in the input file. If columns for coordinates beyond the 'stopsys' coordinate system exist in the input file, then when 'blankcol=yes', their values are replaced with NULL values and when 'blankcol=no', their values are copied to the output file unchanged. If, however, such columns do not already exist in the input file, then they are created and filled with NULL values when 'blankcol=yes', but not created if 'blankcol=no'.
The following seven parameters are used to specify default null value in the processing. Null values may appear for several reasons, including the following: 1. The event time cannot be matched to a row of an attitude file within the time margin 2. The coordinate columns subsequent to the coordinate system specified by 'stopsys' coordinates are set to NULL if 'blankcol' is enabled 3. If an output column already exists in the input file and TNULL for that column is specified in the event file header, then the existing TNULL keyword is copied to the output column. If, however, the output column does not exist in the input file or the column exists, but does not have an associated TNULL keyword, then the appropriate null value parameter is used.

(btnull = 255) [integer]
This parameter sets the null integer value for any output unsigned byte (TFORM = "1B") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(itnull = -999) [integer]
This parameter sets the null integer value for any output short integer (TFORM = "1I") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(jtnull= -999) [integer]
This parameter sets the null integer value for any output long integer (TFORM = "1J") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(ktnull = -999) [integer]
This parameter sets the null integer value for any output long integer (TFORM = "1K") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(sbtnull = 255) [integer]
This parameter sets the null integer value for any output signed byte (TFORM = "1B" with TZERO = -128) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the signed domain.

(uitnull = -999) [integer]
This parameter sets the null integer value for any output unsigned short integer (TFORM = "1I" with TZERO = 32768) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the unsigned domain.

(ujtnull = -999) [integer]
This parameter sets the null integer value for any output unsigned long integer (TFORM = "1J" with TZERO = 2147483648) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the unsigned domain.

(ra = -999.000) [real]
The Right Ascension in decimal degrees of the center of the SKY coordinate images. If ra is outside 0 <= ra <= 360 deg, as with the default value, the ra value is read from the event header of the input file. First the keyword RA_NOM is searched, then RA_PNT. If neither keyword is found, then coordevt exits with an error.

(dec = -999.000) [real]
The declination in decimal degrees of the center of the SKY coordinate images. If ra is outside -90 <= dec <= +90 deg, as with the default value, the dec value is read from the event header of the input file. First the keyword DEC_NOM is searched, then DEC_PNT. If neither keyword is found, then coordevt exits with an error.

(roll = 0.0) [real]
The roll angle about the center of the SKY coordinate system in decimal degrees. The roll angle is the angle measured counterclockwise from Celestial North to the positive SKY Y axis.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Transformation of Hitomi SXI data from RAW to FOC coordinates. For this transformation, there is no need for an attitude file. The teldef file is given specifically in this example and not using CALDB. Note that without specifying stopsys=FOC, the default would be "SKY" and an attitude file would be required.
coordevt infile=ah020150521sxi_a003030201.evt.gz outfile=ah-sxi-coordevt-out.evt /
telfedfile="caldb/data/hitomi/sxi/bcf/teldef/ah_sxi_teldef_20140101v08.fits" stopsys=FOC clobber=yes

2. Transformation of Suzaku data (taking an example from the archive, observation number ae508011020)from RAW to SKY. Users need to specify the format (flags attcol and attform) because the attitude file for this mission is not in the quaternion format (the default for Hitomi) assumed otherwise.
coordevt attfile=ae508011020.att attform=EULER attcol=EULER startsys=RAW stopsys=SKY/
clobber=yes infile=./508011020/xis/event_cl/ae508011020xi0_0_3x3n066l_cl.evt /
outfile=ae508011020 teldeffile=./caldb/data/suzaku/xis/bcf/ae_xi0_teldef_20080303.fits

3. Basic transformation from RAW to SKY coordinates, using the teldef file from the calibration database. The pointing direction is set with the ra, dec, and roll keywords.
coordevt infile="input.fits" outfile="outfile.fits" teldeffile="CALDB" attfile="attitude.fits" ra=0.0 dec=90.0 roll=90.0 clobber="yes"

4. Transformation from ACT to FOC coordinates, using a user-specified teldef file. Since there is no transformation to SKY, an attitude file need not be provided.
coordevt infile="input.fits" outfile="outfile.fits" teldeffile="teldef.fits" startsys=ACT stopsys=FOC clobber="yes"

5. Transformation from SKY to RAW coordinates, using the teldef file from the calibration database. The pointing direction is read from keywords in the infile.
coordevt infile="input.fits" outfile="outfile.fits" teldeffile="CALDB" attfile="attitude.fits" startsys=HIGHEST stopsys=LOWEST \
clobber="yes"

6. Transformation from RAW to SKY coordinates, using the teldef file from the calibration database. The pointing direction is read from keywords in the infile. Floating point columns are output to the outfile. Randomization is disabled and both annual and orbital aberration are enabled. This requires the user to provide an orbit file. The followsun option is also enabled.
coordevt infile="input.fits" outfile="outfile.fits" teldeffile="CALDB" attfile="attitude.fits" orbitfile="orbit.fits" attfile="attitude.fits" \
orbitfile="orbit.fits" inclfloatcol=yes randomize=no annaberration=yes followsun=yes orbaberration=yes clobber=yes

7. Example of a two-stage transformation. In the first pass, coordinates are transformed from RAW to DET, and floating-point values are saved. In the second pass, the floating-point values are used as the starting point to transform from DET to SKY.
coordevt infile="input.fits" outfile="outfile1.fits" teldeffile="CALDB" stopsys=DET /
inclfloatcol=yes clobber=yes coordevt infile="outfile1.fits" outfile="outfile2.fits" clobber=yes teldeffile="CALDB" \
attfile="attitude.fits" startsys=DET inclfloatcol=yes startwithfloat=yes

8. Transformation in which the input files have non-standard formats, extension and column names. Write the output to a user-named log file.
coordevt infile="input.fits" outfile="input.fits" teldeffile="CALDB" attfile="attitude.fits" orbitfile="orbit.fits" clobber="yes" \
annaberration=yes followsun=yes orbaberration=yes infileext="MY_EVENTS" timecol="MY_TIME" attext="GOOD_ATTITUDE" /
attcol="WHATEVER" attform="EULER" orbext="SATELLITE" orbcol="V1,V2,V3" orbform="COMPONENTS" \
logfile="special.log"

9. Transformation using the identity attitude.
coordevt infile="input.fits" outfile="output.fits" teldeffile="CALDB" attfile="IDENTITY" ra=0.0 dec=90.0 roll=90.0

NAME coordpnt - Convert a single point or region file from one coordinate system to another.

USAGE coordpnt input outfile telescop instrume ra dec roll teldeffile startsys stopsys

DESCRIPTION
'coordpnt' is a mission-independent tool for transforming either a single coordinate point or a region file from one coordinate system to another. The input is either a pair of numbers separated by a comma (single point mode) or a filename (region file mode). It converts in both directions (RAW to SKY and SKY to RAW) and through multiple sets of coordinates. See also the help file for the 'coordevt' tool. 'Coordpnt' supports all region shapes supported by the cfitsio 'regfilter' routine. Details about various region shapes can be found at http://ds9.si.edu/doc/ref/region.html.
The regions shapes supported by 'coordpnt' are listed below:
1. Circle: circle x y radius
2. Annulus: annulus x y inner outer n=# / annulus x y r1 r2 r3...
3. Ellipse: ellipse x y radius radius angle
4. Ellipse Annulus: ellipse x y r11 r12 r21 r22 n=# [angle] / ellipse x y r11 r12 r21 r22 r31 r32 ... [angle]
5. Box/Rectangle: box x y width height angle
6. Diamond: diamond point x y
7. PIE/Sector: xcenter ycenter angle1 angle2
8. Polygon: x1 y1 x2 y2 ... xn yn
9. Point: point x y # point=[circle|box|diamond|cross|x|arrow|boxcircle] [size] / circle point x y
10. Line: line x1 y1 x2 y2 # line=[0|1] [0|1]
11. Panda: panda x y startangle stopangle nangle inner outer nradius
12. Epanda: epanda x y startangle stopangle nangle inner outer nradius [angle]
13. Bpanda: bpanda x y startangle stopangle nangle inner outer nradius [angle]
When a region is transformed, all region descriptors (center, dimensions, rotations) are transformed so that in both the destination and origin system, the region covers the same part of the coordinate space. If there are multiple regions within a single input region file, then all regions are transformed to the final coordinate system.

Using the ra/dec/roll and ranom/decnom/rollnom parameters 'coordpnt' may be used to simulate deviations from the nominal pointing to mimic the effects in flight of any imperfections in the attitude control. This is done by adjusting the ranom/decnom/rollnom parameters separately from the ra/dec/roll of the actual pointing. By default, the 'ranom' and 'decnom' are set to the same value that of the pointing, and rollnom is set to 0. These groups of parameters are used only in conversions that include the SKY coordinate system. 'coordpnt' maintains and uses two attitude quaternions. One for the nominal pointing, and the other for the actual pointing. The actual pointing, given by the parameters ra/dec/roll, represents the center of the field of view of the instrument, or the center of the FOC coordinate system. It corresponds to the time-dependent attitude in 'coordevt'. The nominal pointing, given by the parameters ranom/decnom/rollnom, represents the center of the SKY coordinate system and the orientation of the SKY system with respect to the Celestial coordinates. By default the positive SKY-Y axis points toward the North Celestial Pole ('rollnom=0.0').

The transformation between SKY and RADEC coordinates depends only on the nominal pointing ranom/decnom/rollnom and not on ra/dec/roll. The transformation between the focal plane (FOC) coordinates and SKY coordinates depends on the difference between the actual and nominal pointing. If the default parameter settings are used for ranom/decnom/rollnom, then this transformation is only a roll, determined by the 'roll' parameter. If the user chooses to vary 'ranom' and 'decnom' separately from 'ra' and 'dec', then the transformation includes a linear shift in X and Y in addition to a roll. Hitomi specific configuration: Multiseg and winoffset parameters The following parameters multisegpar', 'winoffsetx' and 'winoffsety' are used in the transformation and currently set at default values appropriate for one of the Hitomi/SXI mode. To modify them, users are advised to read the documentation on coordinate system relative to the mission. Below is a short summary for Hitomi.

The 'multisegpar' parameters are used when the input teldeffile contains a 'TRTYPEn = MULTISEG' transformation. This type of transformation involves an initial coordinate system that is multiplexed in some way (e.g. origin X0, Y0 could transform to multiple different destination Xi, Yi). The 'multisegpar' parameters determine the specific transformation to be applied by indexing a MULTISEGn_COEFF table in the teldef file. 'Coordpnt' retrieves the proper coefficients from the table based on the values of the 'multisegpar' parameters and uses these coefficients to build the appropriate transformation based on a hard-coded algebraic formula. This formula also uses the values of two other parameters 'winoffsetx' and 'winoffsety', which are supplied as separate parameters. Below is an example from the Hitomi/SXI.

In the SXI, there are four 'multiseg' properties, which are used to decode the event data in the transformation from the RAW to the ACT coordinates. The properties are listed in the order in which they appear in the SXI TelDef file, which is the order in which the 'multisegpar' parameters must be given.
1. SEGMENT. Each SXI CCD chip has two independent segments, called AB and CD. A photon can land on either segment and the segment ID, 'SEGMENT=0' for segment AB or SEGMENT=1 for segment CD is written to the data stream.
2. READNODE. Each segment has two read-out nodes, A and B for segment AB and C and D for segment CD. The choice of read-out node is commendable in the flight software. The choice of readout nodes is written to the data stream, where 'READNODE=0' denotes node A or C and 'READNODE=1' denotes node B or D.
3. WINOPT. The SXI chips can be operated in commendable "window" modes, wherein a specified part of a chip is read out multiple times during a read out interval. The WINOPT property is set to 'WINOPT=0' if windowing is not used (full chip read out) or to 'WINOPT=1' if windowing is used.
4. WIN_SIZE. There are four possible sizes of the window that can be read out in windowing mode, including full chip readout. The commendable size of the window is written to the data stream with one of four possible values: 'WIN_SIZE=640' for full chip read out, 'WIN_SIZE=160' for 1/4 chip read out, 'WIN_SIZE=80' for 1/8 chip read out and 'WIN_SIZE=40' for 1/16 chip read out. These are not independent of WINOPT; 'WIN_SIZE=640' only applies for 'WINOPT=0', the other values of WIN_SIZE can apply when 'WINOPT=1'. The SEGMENT, READNODE and WIN_SIZE properties are all set independently, yielding a total of 16 (2 x 2 x 4) possible combinations. As an example, to simulate a situation where a photon is read out in segment AB, read node B, with 1/8 windowing mode, one would provide multisegpar parameters "0,1,1,80", for 'SEGMENT=0, READNODE=1, WINOPT=1, WIN_SIZE=80'. Similarly setting multisegpar="1,0,0,640" would mean segment CD, node C, full windowing mode. Any out of range or incompatible parameters leads to an error, like for example multisegpar="2,0,0,640" (out of range) or "1,0,1,640" ('WINOPT' and 'WIN_SIZE' options incompatible).
The 'winoffsetx' and 'winoffsety' parameters are related to the windowing mode, but are set independently. For SXI, 'winoffsetx' should always be zero (it is in fact ignored in the SXI TelDef), and 'winoffsety' can vary, but is nominally linked to WIN_SIZE:
WIN_SIZE=640, winoffsety=1
WIN_SIZE=160, winoffsety=415
WIN_SIZE=80, winoffsety=455
WIN_SIZE=40, winoffsety=475
Currently The Hitomi/SXI is the only instrument to use the MULTISEG capabilities.

PARAMETERS
input [string]
Input point or region to be transformed. The input string can be one of the following.
1) A pair of numbers separated by a comma, representing a single point in the coordinate system specified by the startsys parameter. The default is the lowest-level coordinate system defined in the teldef file.
2) A single pixel number in the case where the startsys coordinate system is represented by pixel numbers, as opposed to positions in a coordinate grid.
3) The name of an ASCII region file containing at least one standard format region defined in the startsys coordinate system.

outfile [filename]
This parameter should be set to "NONE" except when the input parameter is a name of a file. Then it is set to the name of the output file that contains the regions transformed from the startsys to the stopsys coordinate system.

telescop [string]
The name of the telescope for which the coordinate systems in the transformation are defined. The value of telescop must be the name of a valid mission (e.g. Swift, Suzaku, Hitomi) and must match the TELESCOP keyword in the teldef file.

instrume [string]
The name of the instrument for which the coordinate systems in the transformation are defined. The value of instrume must be the name of a valid instrument on the mission given by telescop and must match the INSTRUME keyword in the teldef file.

ra = -999.000 [real]
The Right Ascension in decimal degrees of the actual pointing, representing the center of the field of view of the instrument. This must be a value between 0 and 360.

dec = -999.000 [real]
The declination in decimal degrees of the actual pointing, representing the center of the field of view of the instrument. This must be a value between -90 and +90.

roll = 0.0 [real]
The roll angle in decimal degrees between the SKY coordinate system and the FOC coordinate system. The roll angle is about the center of the SKY system and is measured counterclockwise from the positive SKY Y axis to the positive FOC Y axis.

(ranom = -999.000) [real]
The Right Ascension in decimal degrees of the nominal pointing, representing the center of the SKY coordinate system. When set to -999.000 (the default), 'ranom' is identical to 'ra'. Otherwise, this must be a value between 0 and 360.

(decnom = -999.000) [real]
The declination in decimal degrees of the nominal pointing, representing the center of the SKY coordinate system. When set to -999.000 (the default), 'ranom' is identical to 'dec'. Otherwise, this must be a value between -90 and +90.

(rollnom = 0.0) [real]
The roll angle in decimal degrees between the SKY coordinate system and the Celestial coordinates. The roll angle is about the center of the SKY system and is measured counterclockwise from the North celestial axis to the positive SKY Y axis. By default, 'rollnom' is set to zero, which aligns the SKY system with the Celestial system.

(teldeffile = CALDB) [filename]
Name of the Telescope Definition (TelDef) file, which specifies the coordinate systems and transformation properties. If the parameter is set to CALDB, the default, the file is read from the calibration database. If a file name is provided, then the TELESCOP and INSTRUME keywords in the file must match exactly the values of the telescop and instrume parameters. The tool supports TelDef file format versions 0. through 0.2.

(startsys = LOWEST) [string]
The name of the starting coordinate system of the requested conversion. When startsys is set to 'LOWEST', the default, the conversion begins with the lowest level system present in the teldef file as identified by the parameter 'COORD0', usually 'RAW'. Setting startsys=HIGHEST and stopsys to LOWEST the task converts the top-level coordinates into all the other coordinate systems. In addition to the coordinate systems defined in the teldef file, 'startsys' can be set to 'RADEC', which indicates that the output coordinates are the pair (ra,dec) in decimal degrees.

(stopsys = HIGHEST) [string]
The name of the ending coordinate system of the conversion. If stopsys is set to HIGHEST, then the coordinate transformation chain ends with the highest-level system, usually 'SKY'. In addition to the coordinate systems defined in the teldef file, 'stopsys' can be set to 'RADEC', which indicates that the output coordinates are the pair (ra,dec) in decimal degrees.

(multisegpar = NULL) [string]
The values of the multi-segment parameters, which are required when the teldef file contains a 'TRTYPEn = MULTISEG' transformation (e.g. Hitomi SXI). These parameters must be specified as a list of numbers separated by a comma in with the following way:
a) There must be one number for each parameter. b) The numbers must be in the same order as in the teldef. c) The numbers must be within their allowed range (see the example given above). If there is a MULTISEG transformation for the given instrument, but the multisegpar are left to their default 'NULL' value, the code assumes the property values listed in the first row of the teldef MULTISEGn_COEFF table. If there is no MULTISEG transformation for the given instrument, this parameter is ignored.

(rawtodetseg = 0) [string]
The value of the segment, which is required when the teldef file contains a RAWTODET type transformation governed by segments (e.g. Hitomi SXI), as opposed to a PIXEL_MAP (e.g. Hitomi SXS). This transformation is multiplexed, and the value of 'rawtodetseg' determines the transformation. If the value of this parameter is out of range the code exits with an error.

(pixeltest = CENTER) [string]
This parameter determines how to handle the extent of overlap between a region file and pixels. When pixeltest is set to CENTER (DEFAULT), the center of the pixel must be within the region for inclusion in the output pixel list. When pixeltest is set to PARTIAL, at least one corner of the pixel must be within the region. When pixeltest is set to TOTAL, all corners of the pixel must be within the region. Note that a very small region that does not include any pixel corners may produce a zero-length pixel list.

(winoffsetx = 0.0) [real]
The value of the windowing offset parameter in the X-direction, which is required when the teldef file contains a 'TRTYPEn = MULTISEG' transformation. The value is used, along with the multisegpar parameters to determine the formula for the MULTISEG transformation. Since it is an additive parameter in the transformation formula, when 'winoffsetx=0' (DEFAULT) the parameter is effectively ignored.

(winoffsety = 0.0) [real]
The value of the windowing offset parameter in the Y-direction, which is required when the teldef file contains a 'TRTYPEn = MULTISEG' transformation. The value is used, along with the multisegpar parameters to determine the formula for the MULTISEG transformation. Since it is an additive parameter in the transformation formula, when 'winoffsety=0' (DEFAULT) the parameter is effectively ignored.

(outx = 0.0) [real]
(Output parameter) The value of the output X coordinate when the input is a coordinate point or pixel number. This number is also written to stdout. When 'stopsys' is the lowest level coordinate system and pixel numbers defines that system, 'outx' represents the pixel number and 'outy' is set to zero. This parameter is set to 0.0 (DEFAULT) when the input is a region file.

(outy = 0.0) [real]
(Output parameter) The value of the output Y coordinate when the input is a coordinate point or pixel number. This number is also written to stdout. When 'stopsys' is the lowest level coordinate system and that system is defined by pixel numbers, 'outy' is set to zero and 'outx' represents the pixel number. This parameter is set to 0.0 (DEFAULT) when the input is a region file.

[ccldhm]

EXAMPLES
1. Basic transformation from RAW to SKY coordinates, using the teldef file from the calibration database for the Suzaku XIS0.
coordpnt input="51,51" outfile="NONE" telescop="Suzaku" instrume="XIS0" ra=0.0 dec=90.0 roll=90.0 teldeffile="CALDB"

2. Transformation from RAW to SKY coordinates of a region file, using the teldef file from the calibration database for the Suzaku XIS0.
coordpnt input="region.ds9" outfile="region-out.ds9" telescop="Suzaku" instrume="XIS0" ra=0.0 dec=90.0 roll=90.0 \
teldeffile="CALDB"

3. Transformation of a region file from ACT to FOC coordinates, using a user-specified teldef file for the Swift XRT
coordpnt input="region.ds9" outfile="region-out.ds9" telescop="Swift" instrume="XRT" ra=187.413 dec=-62.998 roll=216.38 \
teldeffile="swxrt-teldef.fits" startsys=ACT stopsys=FOC clobber=yes

4. Transformation from RAW to SKY coordinates for the Hitomi SXS, using the teldef file from the calibration database. The input is a single pixel number.
coordpnt input=15 outfile="NONE" telescop="HITOMI" instrume="SXS" ra=0.0 dec=90.0 roll=90.0 teldeffile="CALDB"

5. Transformation for the Hitomi SXI, where multiseg parameters are set on the command line to specify segment CD, read node C, and 1/8 windowing mode (see Description above for more details on the meanings of these parameters).
coordpnt input="124,56" outfile="NONE" telescop="HITOMI" instrume="SXI" ra=50.0 dec=-32.45 roll=85.88 teldeffile="CALDB" \
multisegpar="1,0,1,80" rawtodetseg=2 winoffsetx=0 winoffsety=455

6. Transformation for the Hitomi SXI, where multiseg parameters are set on the command line to specify segment AB, read node A, and full windowing mode (see Description above for more details on the meanings of these parameters).
coordpnt input="124,56" outfile="NONE" telescop="HITOMI" instrume="SXI" ra=50.0 dec=-32.45 roll=85.88 teldeffile="CALDB" \
multisegpar="0,0,0,640" rawtodetseg=2 winoffsetx=0 winoffsety=1

NAME det2att2 - Converts detector coordinate offsets and rotations to attitude quaternions

USAGE det2att2 infile outfile teldeffile startsys

DESCRIPTION
'det2att2' calculates the spacecraft rotation quaternions corresponding to offsets in detector coordinates. 'det2att2' can also incorporate time-dependent rotations of the detector coordinate system relative to the sky coordinate system. This option requires that the input file has columns for the sine and cosine of the rotation angle; otherwise the rotation angle is assumed to be zero.
'det2att2' is mission-independent and allows the selection of the origin system of the transformation using the parameter 'startsys'.

The quaternion qdelta for an offset (dx, dy) is such that with a mapping of a detector position p = (px, py) to sky coordinates using an attitude q0, yield identical results when applying qdelta * q0 to p as applying q0 to (p - dx, p - dy).

PARAMETERS
infile [filename]
Name of the input offset file.

outfile [string]
Name of the output attitude file.

(teldeffile = CALDB) [string]
Name of the TelDef file. If the parameter is set to CALDB, the file is read from the calibration database.

startsys = SKY_FROM [string]
Name of the starting coordinate system. This must be either one of the coordinate systems defined in the file specified with the 'teldeffile' parameter or the special value 'SKY_FROM' (the default). With TelDef files format versions older than 0.2 and a parameter 'startsys' set to 'SKY_FROM', the originating coordinate system is read from the SKY_FROM keyword in the 'teldeffile.'

(deltaxcol = DELTADETX) [string]
Column name for the offset in the detector X coordinate.

(deltaycol = DELTADETY) [string]
Column name for the offset in the detector Y coordinate.

(sincol = SINANGLE) [string]
Column name for the sine of the rotation angle of the detector coordinates.

(coscol = COSANGLE) [string]
Column name for the cosine of the rotation angle of the detector coordinates.

[ccldhm]

EXAMPLES
1. Convert an offset file derived from the Hitomi CAMS into a delta-attitude quaternion. The input file is the output of 'cams2det.' The TelDef file is that for HXI1, which is the instrument for which the offsets were derived.
det2att2 infile="CAMS_offsets.fits" outfile="delta_attitude.att" teldeffile="ah_hxi1.teldef" startsys=RAW deltaxcol="DELTARAWX" \
deltaycol="DELTARAWY" clobber=yes

2. Convert an offset file derived from the Swift TAM into a delta-attitude quaternion. The TelDef file is that for Swift XRT. The 'deltaxcol' and 'deltaycol' parameters can be the default values. The 'sincol' and 'coscol' parameters are ignored.
det2att2 infile="TAM_offsets.fits" outfile="delta_attitude.att" teldeffile="sw_xrt.teldef" startsys=SKY_FROM clobber=yes

---
NAME eeftable - Create an encircled energy function (EEF) file based on the output history file from the raytracing tool xrtraytrace

USAGE eeftable infile outfile numRadVals deltaRadVals

DESCRIPTION
'eeftable' is a script that creates an EEF FITS file from an output history event file from the tool xrtraytrace. This script allows the user to create EEF files without needing to run the xrtraytrace tool repeatedly. The tool reads each row in the history file, determines which are valid double reflections (meaning it had a single reflection from the primary mirror, a single reflection from the secondary mirror, and hit the results plane). The radii and energy of those photons are then used to calculate the EEF.

INPUT

The input file must be in the same format as output from the tool xrtraytrace, containing information about the event history of every raytraced photon and details of its path. The history file must be large enough to contain at least one valid photon for each radius.

OUTPUT

The output file have an extension containing the EEF for each off-axis angle, azimuthal angle, and energy. Each row is the EEF for a radial value. Off-axis angle, azimuthal angle, and/or energy may be merged in the output file using the 'mgOffaxis', 'mgAzimuth', and 'mgEnergy' parameters. If any of these are set to true, EEF values across that parameter are merged. For example, if mgEnergy=true, there is only one extension for energy instead of one extension for each energy. The EEF values for each off-axis and azimuthal angle is averaged across all the energies. If all three mgOffaxis, mgAzimuth, and mgEnergy are true, then there is a single extension in the output file and all the EEF values are merged into a single extension.
Each extension in the output file contains keywords listing the values used to create the file, such as CBD10000, CBD20000, CBD30000, and OFFAXIS, AZIMUTH, and Energy. If no parameters are merged, those keywords are the exact value of the off-axis angle, azimuthal angle, and energy for that extension. If any of those values of merged, the appropriate keyword reflects the range. The CBDnnnnn keywords contains the range from first unique value to last. The OFFAXIS, AZIMUTH, and Energy keywords are the average of the unique values. The TOTCTS keyword contains the total number of photons that contributed to the EEF for that extension. If any values are merged, the sum of the photons for each parameter is used. For example, if energy is merged, then all the photons that contributed to each energy iteration are summed.

PARAMETERS
infile [filename]
Name of the input photon history file.

outfile [filename]
Name of the desired output file containing EEF for history file.

numRadVals [integer]
The number of radial values to use, which measure the radii of circles centered on the peak of the photon distribution on the focal plane.

deltaRadVals [real]
The size of each radial value, in units of arsec.

(alpha = 0.0) [real]
Exponent to spectrally weigh the energies.

(mgOffaxis = no) [boolean, (yes|no)]
Whether or not off-axis angles should be merged into a single extension.

(mgAzimuth = no) [boolean, (yes|no)]
Whether or not azimuthal angles should be merged into a single extension.

(mgEnergy = no) [boolean, (yes|no)]
Whether or not energies be merged into a single extension.

(cleanup = yes) [boolean, (yes|no)]]
Whether or not to remove temporary files that were created in the process of merging the extensions.

[ccldhm]

EXAMPLES
1. Create an EEF file from a photon history file, without merging.
eeftable.pl in.fits out.fits 100 2.0
Where in.fits is a history file with off-axis angles 0, 15, and 30 arcmin, azimuthal angles of 0 and 45 deg, and energy values of 0.5, 2.0, 3.0, 4.0 keV. The output files contains 24 extensions: an extension containing the radial EEF values for each off-axis/azimuth/energy.

2. Create an EEF file from a photon history file, merging energy.
eeftable.pl in.fits out.fits 100 2.0 mgEnergy=yes
Where in.fits is a history file with off-axis angles 0, 15, and 30 arcmin, azimuthal angles of 0 and 45 deg, and energy values of 0.5, 2.0, 3.0, 4.0 keV. The output file contains 6 extensions: an extension containing the radial EEF values for each off-axis/azimuth.

NAME gticolconv- convert gti file from one extension to multiple extension based on a column value e viceversa

USAGE gticolconv infile outfile column direction keyname keyform

DESCRIPTION
Good Time Interval are defined as file containing START and STOP columns. However if the START and STOP varies with a fix number of elements, a compact format includes beside the START and STOP columns an additional column (hereafter name ELEMENT) containing the element values associate to the appropriate start and stop values.
'gticolconv' has two convertion functions:

1) Split a Good Time Interval (GTI) file containing START STOP and ELEMENT columns into a file containing several extension as many as the unique values in the column ELEMENT .
2) Merge several GTI extensions into a single extension with three columns STATOP STOP and ELEMENT where ELEMENT is populated with a unique value as many as GTI extensions are present.

If the 'direction' parameter is set to SPLIT, 'gticolconv' checks that the GTI extension in the file contains three columns : the columns named START, STOP and the column which named is specified in the parameter 'column'.
The script first calculates the unique values in 'column' and then creates a new GTI extension for each value found in the original input GTI file. The unique value is written in an header keyword specified in the parameter 'keyname'. The format of the value written in 'keyname' is specified in the parameter 'keyform' . The 'keyform' should be specified with a string followed by #s. The script replaces the # 's with the value from the 'column' associated to the GTI specific extension. Similarly, the extension name is set as GTI'keyform' with #'s replaced with column values. The TSTART and TSTOP for each extension are calculated based on the first row START value and last row STOP value for each extension.

If 'direction' is set to MERGE , 'gticolconv' expects a file with several GTI extensions, each containing two columns (START and STOP) and an header keyword specified in the parameter 'keyname' with a pattern specified in the parameter 'keyform'. 'gticolconv' searches for 'keyname' in all GTI extensions and read the # of the 'keyform' to construct the unique values that populate the columns specified in the parameter 'column'. All extensions are merged into a single-extension GTI file with the columns START, STOP and 'column'. The merged extension is named "GTI'column' and the content of the keyword 'keyname' is set as the value of the parameter 'column'. The output file file is then sorted by START, STOP then 'column'. TSTART and TSTOP are calculated based on the first row START value and last row STOP value.

PARAMETERS
infile [filename]
Name of input GTI FITS file. This file may either contain several GTI extensions or one GTI extension.

outfile [filename]
Name of output GTI FITS file.

column [string]
Column name. If the 'direction is split the 'column' parameter is the name of the column in the GTI extension use to split the start and stop in several extensions. If the 'direction' is merge the parameter 'column' is used to name the column containing the values of the elements associated to the start and stop as well as to name the extension and the content of the keyword defined by the parameter 'keyname'

direction [string]
Split GTI or Merge GTI based

keyname [string]
Keyword to read/set output column. If the direction is merge, the input file is search for all the GTI extensions that have in the header the keyword 'keyname'. If the direction is split, the keyword 'keyname' is filled with the parameter value 'column'

keyform [string]
Formatting to read/set for splitting. The content is a string followed by #s. Example PIXEL##

[ccldhm]

NAME gtiinvert - create a gti file by inverting the time intervals of an input gti file

USAGE gtiinvert infile outfile

DESCRIPTION

'gtiinvert' produces a new FITS GTI (Good Time Interval) file where the time intervals are calculated by inverting the time intervals of an input GTI file. 'gtiinvert' operates on a FITS GTI file, where the times of the intervals are written in the START and STOP columns. The times are seconds from a start time written in the header MJDREF/FI keywords. By default the 1st interval in the output GTI file starts with the first gap in the input GTI file, e.g. the 1st start time in the START column of the output GTI corresponds to the 1st interval stop time of the input GTI. Similar the last interval in the output GTI file correspond to the last gap of the input GTI file. e.g. the last stop time in the STOP column of the output GTI correspond to the last interval start time in the input file. This is the 'gtiinvert' behavior when the parameters 'tstart' and 'tstop' set to 'DEFAULT'. By setting the parameters 'tstart' and/or 'tstop' to values different than 'DEFAULT' and 'margingti' to 'yes', 'gtiinvert' uses these values to set the 1st interval start time and/or the last interval stop time in the output GTI file respectively. The parameter 'dt' defines a delta time around each interval in the output GTI file as show below.
INPUT GTI: ----------------------| |--------------------------------
OUTPUT GTI: |dt|------------------|dt|

The output GTI file has only two columns , START and STOP and the extension name may be specified with the parameter 'outext'.

PARAMETERS
infile [filename]
Name of input GTI FITS file.

outfile [filename]
Name of output GTI FITS file.

outext [string]
Name of extension in output GTI FITS file.

(margingti = yes) [boolean]
Create the first and last GTI based on the TSTART/TSTOP specified by the parameters 'tstart' and 'tstop'.

(tstart = DEFAULT) [string]
Value to use for TSTART in seconds. If set to DEFAULT, then the value is taken from the TSTART keyword of the infile.

(tstop = DEFAULT) [string]
Value to use for TSTOP in seconds. If set to DEFAULT, then the value is taken from the TSTOP keyword of the infile.

(dt = 0) [real]
Delta time added to all GTI start times and subtracted from all GTI stop times in the output GTI (seconds).

[ccldhm]

Mode of automatic PARAMETERS
EXAMPLES
Invert a GTI file named test_gti_in.fits and named the output GTI extension GTIOUT
gtiinvert infile=test_gti_in.fits outfile=test_gti_out.fits outext=GTIOUT

NAME gticolconv -- convert GTI file from one extension to multiple extension based on a column value and viceversa

USAGE gticolconv infile outfile column direction keyname keyform

DESCRIPTION
Good Time Interval are defined as file containing START and STOP columns. However if the START and STOP varies with a fix number of elements, a compact format includes beside the START and STOP columns an additional column (hereafter name ELEMENT) containing the element values associate to the appropriate start and stop values.
'gticolconv' has two convertion functions:
1) Split a Good Time Interval (GTI) file containing START STOP and ELEMENT columns into a file containing several extension as many as the unique values in the column ELEMENT .
2) Merge several GTI extensions into a single extension with three columns STATOP STOP and ELEMENT where ELEMENT is populated with a unique value as many as GTI extensions are present.

If the 'direction' parameter is set to SPLIT,'gticolconv' checks that the GTI extension in the file contains three columns : the columns named START, STOP and the column which named is specified in the parameter 'column'. The script first calculates the unique values in 'column' and then creates a new GTI extension for each value found in the original input GTI file. The unique value is written in an header keyword specified in the parameter 'keyname'. The format of the value written in ‘keyname’ is specified in the parameter

'keyform' . The 'keyform' should be specified with a string followed by #s. The script replaces any #'s with the value from the 'column' associated to the GTI specific extension. Similarly, the extension name is set as GTI'keyform' with #'s replaced with column values.
The TSTART and TSTOP for each extension are calculated based on the first row START value and last row STOP value for each extension.

If 'direction' is set to MERGE, 'gticolconv' expects a file with several GTI extensions, each containing two columns (START and STOP) and an header keyword specified in the parameter 'keyname' with a pattern specified in the parameter 'keyform'. 'gticolconv' searches for 'keyname' in all GTI extensions and read the # of the 'keyform' to construct the unique values that populate the columns specified in the parameter 'column'. All extension are merged into a single-extension GTI file with the columns START, STOP and 'column'. The merged extension is named "GTI'column" and the content of the keyword 'keyname' is set as the value in the parameter 'column'. The output file is sorted by START, STOP and then by 'column'. TSTART and TSTOP are calculated based on the first row START value and last row STOP value.

PARAMETERS

infile [file]
Input GTI file. This file may either contain several GTI extensions or one GTI extension.

outfile [file]
Output GTI file

column [string]
Column name. If the 'direction is split the 'column' parameter is the name of the column in the GTI extension use to split the start and stop in several extensions. If the 'direction' is merge the parameter 'column' is used to name the column containing the values of the elements associated to the start and stop as well as to name the extension and the content of the keyword defined by the parameter 'keyname'

direction [string]
Split GTI or Merge GTI based

keyname [string]
Keyword to read/set output column. If the direction is merge, the input file is search for all the GTI extensions that have in the header the keyword 'keyname'. If the direction is split, the keyword 'keyname' is filled with the parameter value 'column'.

keyform [string]
Formatting to read/set for splitting. The content is a string followed by #s. Example PIXEL##

EXAMPLES
Splitting a file:
gticolconv ah100041010sxs_el.gti.gz[1] px_split.gti PIXEL SPLIT DETNAM
PIX## clobber=yes
Merging a file:
gticolconv px_split.gti px_merge.gti PIXEL merge DETNAM PIX##
clobber=yes

NAME heasim -- multi-mission high-energy astrophysics simulation tool

USAGE heasim mission instrume rapoint decpoint roll exposure insrcdeffile outfile psffile vigfile rmffile arffille intbackfile

DESCRIPTION
Heasim is a multi-mission high-energy astrophysics simulation tool. Heasim utilizes a pseudo-Monte-Carlo approach that redistributes the source photons in position and energy according to the input files appropriate to the chosen telescope/detector system. The output is a FITS event file, from which data products may be extracted and analyzed using, e.g., XSELECT, XSPEC, etc.

The parameters mission and instrume are for specifying the mission name and the instrument names for that mission. The specific characteristics of the supported instruments (e.g., pixel size, field-of-view, etc.) are collected in the mission database file (parameter mdbfile). The parameters rapoint, decpoint define the pointing direction, and the roll parameter the rotation angle (all in decimal degrees). The exposure time in seconds is entered in as the parameter exposure. The source(s) position; and spectral, spatial, and temporal characteristics are input via the source definition file (parameter insrcdeffile). Point spread function, vignetting, spectral response, effective area and internal background for each detector are input using the parameters psffile, vigfile, rmffile, arffille, and intbackfile, respectively. The sky background may be calculated using the skyback tool and the output files from skyback may be input

to heasim. The parameter psbackfile accepts a file containing a list of the resolved background point sources as calculated and output by skyback. The parameter pszbackfile accepts a file that lists the point source intrinsic redshifts and absorptions for these same sources. The parameter difbackfile accepts the file that lists a single diffuse background extended source as calculated and output by skyback. The simulation may be split into multiple subexposures by setting the flagsubex parameter to yes, and the subexpsosure parameter to a value that differs from exposure. Resampling of event locations from sky pixels onto larger detector pixels is controlled by the resample parameter. Basic event pileup may be simulated with the dtpileup timescale parameter in units of seconds.

Heasim calculates simulated events for each input source in the source definition file (and background) with the following steps. (1) The absorbed input spectrum (in photons cm-2 s-1 vs keV) based on the input spectral model or user spectrum file is calculated in each energy bin in the arf file. (2) This is then converted to cts/bin vs bin on this input energy grid by multiplying by the exposure time and effective area. (3) The sum over all bins yields the total number of events. For each source, the code loops over each input energy bin, and over the counts in each bin (4) probabilistically assigning a sky position to each event according to the source spatial distribution. This distribution may be determined either by a model or by an input image (or subimage). (5) An event may be discarded based on the vignetting, and (6) "scattered" by the PSF. (7) Events are discarded if located outside the FOV, and (8) assigned final discrete sky coordinates that are resampled within detector pixels unless resampling is switched off. For each energy bin, (9) the rmf is used to redistribute the input energy and assign output PI values for each photon not discarded due to vignetting or falling outside the FOV. These events are then (10) assigned times according to the source temporal characteristics (constant, periodic, or burst). Sky background sources, i.e. from the output of skyback, are identically processed. Internal background events from the input internal background spectrum, following rescaling to the detector area and observation exposure time, are assigned random detector locations and times within the exposure. Events from all sources and from the internal background are merged and sorted on time, with an option to identify pileup. If the simulation is split into subexpsoures, this procedure is repeated for each subexposure, with the time-sorting, pileup calculation, and final output completed within each interval to reduce memory usage.

The "source definition file" allows one to specify the spatial, spectral, and temporal characteristics of the source(s) to be simulated. This file is always required. This is an ascii file where each row contains parameters, separated by commas, and is dedicated to a single source. The format of each row is: ra, dec, NH, spec_mod, spec_par, flux, bandpass, filename, format, units, source_specs

where ra and dec identifies the position of the source to simulate (degrees), and NH is the column density. The spectral characteristics may be specified in two different ways: (a) as a single component spectral model with one spectral parameter, and flux in a specified energy band, using the parameters spec_mod, spec_par, flux, and bandpass; or, alternatively (b) a spectrum can be input using the parameters filename, format, and units (see description in the user guide). All of these parameters are required, except for source_specs -- a string that may be added to specify the timing or spatial characteristics for sources. The specific strings are: pulse(period, pulse_frac); burst(tburst, risetime, decaytime, burstratio); extmod(beta, beta, core_radius, ellipticity, pos_angle, Rmin, Rmax); extmod(ellipse, ellipticity, pos_angle, Rmin, Rmax); extmod(power, slope, Rmin, Rmax); extmod(gauss, fwhm_x, fwhm_y, pos_angle); extmod(flat, Rmin, Rmax); image(filename, xmin, xmax, ymin, ymax). These options are described in detail in the heasim user guide.

Setting the flagsubex parameter to "yes" and the subexposure parameter to a value different than the exposure parameter may be used to reduce memory usage by splitting the simulation into several separate, consecutive sub-simulations. In this case heasim processes the simulated events within the time specified in the subexposure, writes these events to the output file and proceeds to the next subexposure. If the subexposure is equal to the exposure parameter all events are processed together (even for flagsubex=yes). If subexposure > exposure, the subexposure durations are "optimized" to have less than 500,000 counts per subexposure. By default, the parameters flagsubex and subexposure are set to allow the "optimization" mode (flagsubex=yes and subexposure =1.0e9). This option cannot be used to simulate variable sources.

If the skipfov parameter is set to "no", events outside of the field-of-view as defined by the mdb file are not discarded.

The heasim output events file contains the following columns: the time of the event, TIME, the sky coordinates, X and Y, the energy channel PI, and a flag set to 1 if the event is piled up (and 0 otherwise), PILEUP.
For additional details on the parameters, source definition file format and options, the mission database content, calibration file format, and on running heasim please see the heasim user guide.

PARAMETERS
mission [string]
Name of the mission to be simulated, i.e. Hitomi, Suzaku, XMM. Required.

instrume [string]
Name of the instrument to be simulated, i.e. SXS, SXI, HXI, etc. Required.

rapoint [double]
Right Ascension of the pointing position. This may be different from the source right ascension specified in the source definition file. The units are decimal degrees in J2000 epoch. Required.

decpoint [double]
Declination of the pointing position. This may be different from the source declination specified in the source definition file. The units are decimal degrees in J2000 epoch. Required.

roll [double]
Rotation angle applied to the y-axis when converting between RA/DEC and coordinates aligned with the detector, i.e. the detector is rotated by the roll angle from north in the sky. Given in decimal degrees. Required.

exposure [double]
Exposure time in seconds (i.e. the observation duration). Longer simulation times result in more photons being "detected." Required.

insrcdeffile [string]
The filename of the source definition file (ASCII) with the source(s) characteristics to simulate. Required.

outfile [string]
The filename of the simulated output FITS event file. Required.

psffile [string]
The filename with the point-spread function (PSF) information - either in PSF or EEF form - for use in the simulation. The file format is either ASCII or FITS (see the calibration files section of the heasim users guide). If set to "Gaussian", a Gaussian psf with FWHM from the mission data base file is used. If set to NONE the photons are not scattered and maintain the original source distribution. Required.

vigfile [string]
The filename with the vignetting information. The file format is either ASCII or FITS (see the calibration files section of the heasim users guide). If set to NONE no vignetting is applied. Required.

rmffile [string]
The filename with the response matrix file (rmf) information. The rmffile is used to assign energy bin to events (PI value), since for a given input energy it gives the probability redistribution of the input energy into the final output energy grid. If the parameter is set to 'NONE', the final energy bin assigned to the event do not use the redistribution matrix but instead the energy grid from the arf.
The file format is FITS (see the calibration files section of the heasim users guide). Required.

arffile [string]
The filename with the ancillary response file (arf) information. This file contains the effective area for different energies. The file format is FITS (see the calibration files section of the heasim users guide). Required.

intbackfile [string]
The filename with the internal background spectrum of the instrument. The file format is a standard spectral file in FITS format FITS (see the calibration files section of the heasim users guide). If set to NONE, heasim does not add events due to the internal background. Required.

(psbackfile = NONE) [string]
The filename for the sky background component from resolved background point sources. The file contains, for each source, the sky position and spectral information. These resolved background point sources may be calculated by skyback. The format is similar to the source definition file. Heasim calculates the events from the background point sources and adds them to the final event file. Optional.

(difbackfile = NONE) [string]
The filename for the unresolved sky background contribution. The file contains data for the central sky position, radial extent, and name of file for the diffuse spectrum. Both this source file and spectrum for the unresolved background may be calculated by skyback. The format is similar to the source definition file. Heasim calculates the events from the unresolved background and adds them to the final event file. Optional.

(pszbackfile = NONE) [string]
The filename with the auxiliary data of the resolved background point sources. The file contains redshift and intrinsic absorption column density of the resolved background point sources in psbackfile. The format is two columns, with redshift in the first, and intrinsic column density in the second. These may be calculated by skyback. If this file is absent, all redshifts and intrinsic absorption column densities are assumed to be 0. Optional.

(arfrmftol = 1.0) [double]
Tolerance required for the agreement of the rmf and arf energy scales. If the tolerance value exceeds the set value, the simulation exits prematurely warning the user. Optional.

(flagsubex = yes) [boolean]
If set to yes, the simulation is split into multiple observations. This may be used to reduce memory usage (only) for sources that do not vary in time. By default, the parameters flagsubex and subexposure are set to allow the "optimization" mode (flagsubex=yes and subexposure =1.0e9) ([yes]/no). Optional.

(subexposure = 1.0e9) [double]
Subexposure time in seconds. If flagsubex is set to yes, this parameter is used to effectively split the simulation into multiple observations with this duration. If greater than the exposure parameter, the subexposure duration is "optimized" such that there are no more than 500,000 estimated counts per subexspoure. By default, the parameters flagsubex and subexposure are set to allow the "optimization" mode (flagsubex=yes and subexposure =1.0e9). Optional.

(resample = yes) [boolean]
If set to yes, the coordinates of simulated photons are recalculated by placing them at a random location within a detector pixel ([yes]/no). Optional.

(skipfov = no) [boolean]
Skip discarding of out-of FOV events (yes/[no]).

(dtpileup = 0.0) [double]
Timescale for pileup in seconds. Two or more events are flagged if their times are within this timescale and in the same detector pixel. These events are flagged as "pileup" events. Optional.

(getinfile = no) [boolean]
If set to yes, heasim_source_sample.txt is copied to the current working directory, and used in place of whatever the source_input_filename parameter was set to. In addition, the pointing determined by the rapoint and decpoint parameters is replaced by the source position (yes/[no]). Optional.

(filter = NONE) [string]
NOT CURRENTLY USED, set to "NONE". Specifies the instrument filter used.

(instmode = NONE) [string]
NOT CURRENTLY USED, set to "NONE". Specifies for example the CCD readout mode.

(seed = 1) [integer]
Value with which to seed the simulator's random number generator (RNG). If set to zero, the seed is ignored and the RNG is seeded from the system time. Otherwise it is used as given and consecutive applications with identical seed parameter yield identical output.

(mdbfile="$ENV{LHEA_DATA}/heasim.mdb") [string]
The filename of the mission database file, which gives mission-specific parameters needed for the simulated observation (see the MDB file section). The default heasim.mdb file may be found under $HEADAS/..//refdata.

(clobber = no) [boolean]
Overwrite the existing output file if set to yes (yes/[no]).

(debug = no) [boolean]
Diagnostic output is printed out on the screen if set to yes (yes/[no]).

(mode = ql) [string]
Mode to query the parameter file. Acceptable values include: "ql (query and learn/remember), "hl" (hidden and learn/remember), "q" (query but don't remember), "h" (hidden).

EXAMPLES

1. Simulate 10 ksec Hitomi SXI simulation of a source characterized by the source definition file "source.txt", with pointing direction ra=150, dec=50. The input calibration EEF, rmf, and arf files are sxi_eef.fits, sxi.rmf, and sxi.arf; and there is no vignetting, internal background, or sky background in the simulation. The subexposure option is turned off, and the output events fits file is hitomi_sxi.fits.
heasim hitomi sxi 150.00 50.00 0.0 10000. source.txt hitomi_sxi.fits sxi_eef.fits none sxi.rmf sxi.arf none flagsubex=no

---

NAME hitomiversion - report version string for headas/hitomi package

USAGE hitomiversion

DESCRIPTION
This tool reports the version string for the headas/hitomi package. The version information is contained in the hitomiversion Perl script itself.

PARAMETERS
none

NAME hxievtid -- Reconstructs HXI events

USAGE hxievtid infile outfile remapfile fluorefile badpixfile enecutfile

DESCRIPTION
'hxievtid' determines whether the signals in an occurrence constitute a valid event and computes the event position and the energy deposited in the detector.

The input file to 'hxievtid' is an SFF file with variable length array columns where each row contains an occurrence with multiple signals recorded at a given time. The output event file, specified by the parameter 'outfile', has a different structure and do not contain columns with variable length arrays.

In this output file, each row corresponds to a reconstructed event and the columns 'RAWX', 'RAWY' and 'PI' contain the reconstructed position and energy of that event along with additional columns containing other event characteristics, as described below. The HXI consists of five detector layers in a vertical stack: four Si strip detectors at the top and a single CdTe detector at the bottom. Each layer is a strip detector with topside strips running perpendicular to bottom side strips. Signals in more than one layer are rejected unless they are reconstructed with an X-ray fluorescence event. Signals below the threshold PHA values given in badpixfile are not used in the reconstruction.

The only localization patterns recognized as a valid signal are:

1. One signal on the top side and one signal on the bottom side.
2. One signal on the top side and two signals in adjacent strips on the bottom side.
3. Two signals in adjacent strips on the top side and one signal on the bottom side.
4. Two signals in adjacent strips on the top side and two signals in adjacent strips on the bottom side.

Occurrences with signals matching one of the template patterns in both the CdTe layer and one Si layer may be accepted, if the reconstructed energy deposited in the Si layer is within one of the ranges for a fluorescence X-ray from the CdTe layer.

The reconstructed energy of an event is computed using the signal EPI values calculated in hxisgdpha. For events occurring in a single layer, the event energy (EPI) in keV is taken to be the sum of the one or two signals in the top layer for Si or the bottom layer for CdTe. For events with signals in a single Si layer and the CdTe layer, the EPI is taken from the one or two bottom layer signals in the CdTe layer. The PI value of the event is calculated using a channel size of 0.1 keV.

The coordinate location (RAWX, RAWY) is evaluated by taking the strip number associated with the signal with the larger EPI in both top (RAWY) and bottom (RAWX) sides. In the case of an occurrence with both signals equally split and having the same EPI, then the signal determining the position is chosen randomly. For events with signals in two layers, one on Si and one on CdTe, the position this procedure is applied for signals in the CdTe layer.
Reconstruction results for each occurrence are in the output file specified by the 'outfile' parameter.
Columns copied from the input SFF are TIME, OCCURRENCE_ID, CATEGORY,
FLAG_SEU, FLAG_LCHK, FLAG_TRIG, FLAG_TRIGPAT, FLAG_HITPAT, FLAG_FASTBGO, LIVETIME, PROC_STATUS, and STATUS.
Output file columns populated by the reconstruction are as follows:
EVTCAT: event category = 1 for 'one hit, good', 2-5 for 'two hits, good', 6 for 'energy test failed', 7 for 'two hits, both Si layer', 8 for 'invalid hit or hits', 9 for 'no hits found', 10 for 'reconstruction failed, more than two hits, 11 for 'no reconstruction, output set to NULL'
SIGNAL: number of signals per side
SIGPOS: signal position per side (X or Y)
GOODBAD: good (0) or bad (1) flag per side
VALIDHITS: hit validity per layer
LAYER: layer number = 0-3 for Si or 4 for CdTe
ENE_TOTAL: sum of EPI per occurrence
PI: pulse invariant
RECO_STATUS: bit flags describing the reconstruction of each occurrence (see below)
RAWX, RAWY: HXI RAW pixel coordinates of each reconstructed event

EPITOP: total EPI for top layers
EPIBOT: total EPI for bottom layers
EPICUT: code for energy cut = 0 for OK, 1 for 'exclude high', 2 for 'exclude low', 3 for 'no signal', 4 for 'below threshold', or -999 as a null value. Additional coordinate columns are added to the file but populated later by task 'coordevt': ACTX, ACTY, DETX, DETY, FOCX, FOCY, X, and Y.
The bit flags in the RECO_STATUS (reconstruction status) column are as follows:
Bit Description
------(Occurrence mode identification)----------
0 Mode is Am 241
1 Mode is PSEUDO
2 Mode is CALMODE
3 Mode is READALL
4 Mode is NOTSURE
------(Event reconstruction skipped for this occurrence)----------
5 Bad PROC_STATUS
6 Mode is READALL or CALMODE and SKIPRECO option is YES
7 Occurrence has no signals
8 FASTBGO or HITPAT flag is set and REJECTBGO option is YES
------(Event reconstruction outcome)----------
9 Event could not be reconstructed

PARAMETERS
infile [filename]
Input HXI event date file. This is the SFF, output of the hxisgdpha task.

outfile [filename]
Output event file.

remapfile = CALDB [filename]
File containing the remapping of ASIC and READOUT numbers. If the parameter is set to CALDB, the default, the file is read from the calibration database.

fluorefile = CALDB [filename]
File containing the energy ranges of fluorescence electrons in CdTe and Si. If the parameter is set to CALDB, the default, the file is read from the calibration database.

badpixfile = CALDB [filename]
File containing the energy threshold defining a real signal for each readout. If the parameter is set to CALDB, the default, the file is read from the calibration database.

enecutfile = CALDB [filename]
File containing criteria for rejecting events with a large difference between the top- and bottom-side energies in a layer If the parameter is set to CALDB, the default, the file is read from the calibration database.

(occurrenceid = -1) [integer]
If greater than zero, the task only expands this single occurrence.

(rejectbgo = no) [boolean]
Reject events (the default) for which BGO trigger occurred. (yes/[no])

(outcalfile = NONE) [filename]
Output a file containing the reconstruction information for each occurrence by detector side. This file is used is used for calibration and not useful for science analysis. The default is to set this parameter to NONE.

(datamode = NONE) [string]
Substitute DATAMODE in place of event value (or NONE)

(skipreco = no) [boolean]
By default skip the reconstruction of READALL and CALMODE events.(yes/[no])

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Reconstruct HXI events using the default values of all hidden parameters. This option produces no extra output file.
hxievtid infile=HXI_SFF.fits outfile=HXI_output.fits remapfile=CALDB fluorefile=CALDB badpixfile=CALDB enecutfile=CALDB

2. Reconstruct HXI events and produce an extra output file.
hxievtid infile=HXI_SFF.fits outfile=HXI_output.fits remapfile=CALDB fluorefile=CALDB badpixfile=CALDB enecutfile=CALDB \
outcalfile="extra.fits" clobber=yes

3. Reconstruct HXI events for a single occurrence.
hxievtid infile=HXI_SFF.fits outfile=HXI_output.fits remapfile=CALDB fluorefile=CALDB badpixfile=CALDB enecutfile=CALDB \
occurrenceid=2 clobber=yes

NAME hxigainfit -- Calculate the HXI time-dependent energy gain corrections for events from comparison with known calibration lines

USAGE hxigainfit infile outevtsuffix outfile

DESCRIPTION
'hxigainfit' is a script that runs 'ahgainfit' on HXI event files and allows the user to select the events input into 'ahgainft'. The description of 'ahgainfit' and its parameters also used by 'hxigainfit' are described below.

The HXI gain is calculated using reconstructed AM241 event files as input files. 'hxigainfit' calculates the time dependent shift of the calibration source energy. 'hxigainfit' includes options to (a) create gti (see parameter 'rungtigen'), (b) screen the events ('runscreen'), prior to running 'ahgainfit'.

(a) If 'rungtitgen=yes', 'hxigainfit' creates a GTI file. The GTI are created using the expression either specified in the parameter 'gtiexpr' and/or using the default expression label ('gtigenlabel' parameter) stored in a CALDB file ('selectfile' parameter). The expression label refers to entries listed in the input mkf file ('gtigen_infile' parameter). If the 'gtifile' parameter is set, the single GTI file or a list of GTI files specified is also used.

(b) If 'runscreen=yes', 'hxigainfit' screens the input event data with the expression specified in the parameter 'expr' and/or using the default expression label ('screenlabel' parameter) stored in the CALDB file specified by the 'selectfile' parameter.

'hxigainfit' creates two outputs. The first is a gain correction file identical in structure to that produced by 'ahgainfit'. The second is the screened event file created within 'hxigainfit'. The filename of the screened event file is the same as the input file with a suffix (see 'outevtsuffix' parameter) appended.

ahgainfit:

'ahgainfit' calculates time-dependent energy gain corrections by comparing the theoretical and observed energies of a calibration line or line complex. For each run of the task, only one line may be specified to calculate the gain correction (see parameter 'linetocorrect'). 'ahgainfit' is a general task that is used directly in the scripts 'hxigainfit', 'hxigainfit', and ‘sgdgainfit’ for the SXI HXI, and SGD respectively. For the SXS, the 'sxsgain' tasks uses the same core fitting function of 'ahgainfit'.

'ahgainfit' takes as input an event file with time and energy columns, and requires that the events are time ordered. The task accumulates spectra from events that are consecutive in time, with energy centered on the calibration feature and compares each spectrum from the events with a theoretical model of the calibration feature profile. The calibration feature used by 'ahgainfit' is specified by the parameter 'linetocorrect' as a string, and the names and energies of the features are specified in a calibration file (parameter 'linefitfile'). The calibration feature may be composed of many atomic or nuclear line components that are listed in the calibration file.

The energy range for the spectra constructed from the event file as well as from the theoretical profile, may be specified in two different ways. (1) The default energy range for the spectra is the smallest and largest energies of the line components, and expanded with the 'extraspread' parameter, i.e. [E_min - extraspread : E_max + extraspread]. It is recommended to set 'extraspread' larger than the sum of the natural width of the calibration feature, the value of the 'broadening' parameter, and the magnitude of the expected energy shift. (2) Alternatively, the energy range may be specified by setting the 'startenergy' and 'stopenergy' parameters. If these parameters are non-negative 'ahgainfit' uses their values to accumulate the spectra, instead of the range derived using the 'extraspread' setting. The energy column used to accumulate the spectra is specified by the 'energycol' parameter. The energy column is expected to be in units of channel, where the eV per channel is set by the 'evchannel' parameter. The spectra are binned according to the 'binwidth' parameter, where the

'binwidth' is in units of 'energycol' channels. The theoretical profile is constructed on a mesh defined by this energy range and 'binwidth', where each calibration line is assumed to be Lorentzian. The profile may be convolved with a Gaussian having the FWHM given by the 'broadening' parameter.

The number of events in each spectrum is defined by the 'numevent' and 'minevent' parameters. The task accumulates spectra with a number of events between 'minevent' and 'numevent'. However, if a spectrum has fewer than 'minevent' points, then it is combined with the previous spectrum if possible. Therefore all spectra have a size between 'minevent' and ('numevent+minevent-1'). To avoid having spectra accumulated over large gaps in time, the group of points in the spectrum is truncated when the time interval between consecutive events is greater than the 'gapdt' parameter. Adjacent spectra in time may share a percentage of their points based on the 'grpoverlap' parameter that may vary between 0 and 100. If 'grpoverlap' is set to 0, the consecutive spectra share no points in common; if set to 100 they share all points in common but one.

The spectra events may be simultaneously collected based on the value of a column present in the event file, given by the 'splitcol' parameter. This option may be used, for example, to find the gain correction for each layer of the HXI detector ('splitcol=LAYER'). If a GTI file is specified by the 'gtifile' parameter, events outside of these GTI intervals are excluded. Spectra are not accumulated across GTI intervals unless the 'spangti' parameter is set to 'yes'.

For each accumulated spectrum, 'ahgainfit' fits the theoretical profile to the data and also derives binned and unbinned averages. A least-squares method is used in the fitting. The fitted parameters are energy shift, scaling factor, background (unless the 'background' parameter is set to NONE), and, optionally, convolution width if 'fitwidth=yes'. The background is fit with a constant value if 'background' is set to CONSTANT, and a power-law if set to SLOPE. The unbinned average energy (as specified by 'energycol') is the sum of the energies in the spectrum divided by the number of events in the spectrum. The binned average energy is the weighted average derived by summing, over bins in the spectrum, the product of the energy and number of events per bin, and then dividing by the total number of events in the spectrum. The fitted gain corrections is computed from the fitted shift with respect to the theoretical line profile. The binned average gain correction is computed from the difference between the profile and spectrum averages.

The default values for the parameters used in the fitting method ('minwidth0', 'maxitcycle', 'r2tol', 'searchstepshift', 'maxdshift', 'bisectolshift', 'searchstepwidth', 'maxdwidth', 'bisectolwidth', and 'minwidth') should not need to change since already optimized.

The output file has two extensions. One extension, GRID_PROFILE, contains the energies and amplitudes of the theoretical profile used in the fitting procedure, including any convolution from the 'broadening' parameter. The other extension, DRIFT_ENERGY, reports the fitting results for each spectra in the following columns: TIME (midpoint of the time interval over which the spectrum is collected), splitcol (value of splitcol for spectrum as given by the 'splitcol' parameter; this column is absent if 'splitcol=NONE'), COR_FIT (energy correction factor from spectrum fit), COR_AVE (energy correction factor from spectrum average), CHISQ (reduced chi-squared of the fit), AVGUNBIN (average energy of events in spectrum prior to binning), AVGBIN (weighted spectrum average energy), AVGFIT (average energy from fit), SHIFT (fitted energy shift), SCALE (fitted vertical scaling factor), BGRND (fitted background), WIDTH (if 'fitwidth=no', same as broadening parameter; if 'fitwidth=yes', fitted width), TELAPSE (difference between times of first and last event in spectrum), EXPOSURE (calculated using the GTI), NEVENT (total number of events collected for this spectrum), BINMESH (array containing the count spectrum energy bins), SPECTRUM (array containing the observed binned count spectrum), FITPROF (array containing theoretical profile with fitted parameters applied). If the 'calcerr' parameter is set to 'yes', one-sigma errors for the SHIFT and WIDTH are calculated. The errors are calculated with chi-squared and maximum-likelihood methods and output in the columns SIGSHCHI2, SIGWDCHI2, SIGWDLIKE and SIGWDLIKE respectively. If 'writeerrfunc' parameter is set, the chi-squared and likelihood calculated valued are output in the arrays SHCHI2, SHLIKE, WDCHI2 and WDLIKE. The numbers of values output in these arrays are specified in the 'nerrshift' and 'nerrwidth' parameters, respectively.

PARAMETERS
infile [filename]
Input event file name. Input file may be a single file or a list of files (the name of the text file with the list preceded by the "@" character).

outevtsuffix [string]
Screened output file name suffix. This string is appended to the input file name (or name of first input file if a list).

outfile [filename]
Output gain correction file name.

(rungtigen = no) [boolean]
If set to yes, run ahgtigen to create GTI (yes/[no]).

(runscreen = no) [boolean]
If set to yes, run ahscreen to filter input file (yes/[no]).

(gtigen_infile = mkf.fits) [filename]
Input mkf file used to create a GTI used in screening. This parameter is ignored if rungtigen is set to no.

(gtifile = NONE) [string]
Input gti file, or name of text file with list of files, preceded by "@" if including multiple input gti files to be merged. If the parameter is set to NONE no input GTI file is used. This parameter is ignored if rungtigen is set to no.

(selectfile = NONE) [string]
CALDB or user input label file with labels and expressions. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used.

(gtigenlabel = NONE) [string]
Labels to read from label file specified by selectfile and applied to gtigen_infile to create GTI, or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if rungtigen is set to no.

(screenlabel = NONE) [string]
Labels to read from label file specified by selectfile and used to screen input event file(s), or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if runscreen is set to no.

(gtiexpr = NONE) [string]
Expression, or label to read from label file specified by selectfile, used to create GTI. If the parameter is set to NONE no expression is used. This parameter is ignored if rungtigen is set to no.

(expr = NONE) [string]
Additional event screening expression used to screen input event file(s). If the parameter is set to NONE no expression is used. This parameter is ignored if runscreen is set to no.

(linefitfile = line.in) [filename]
Input CALDB file containing parameters of the Lorentzian components used to construct the theoretical line profile. If the parameter is set to CALDB, the file is read from the calibration database.

(linetocorrect = MnK) [string]
Calibration line to use for the gain correction. The value must match an extension in linefitfile. For the HXI the strongest lines are the 13.93 keV NpLa and 17.51 keV NpLb lines in the Si layers, and the 59.5412 Am60keV for the CdTe (top) layer.

(energycol = PI) [string]
Name of energy column to use in gain fitting.

(splitcol = SIDE) [string]
Column name used to separate the data.

(numevent = 1000) [integer]
Number of events collected for each spectrum used to calculate a single gain correction.

(minevent = 750) [integer]
Minimum number of events required for a spectrum. If the length of a group is less than minevent, those points are included with the previous group for processing, if possible.

(gapdt = -1.) [real]
The upper limit to the time interval between two consecutive events in the same spectrum used in fitting. Two consecutive events separated in time by more than this amount are assigned to different groups. If gapdt=-1, no limit is imposed.

(grpoverlap = 0.) [real]
The percentage overlap between adjacent groups. For grpoverlap=100 adjacent groups are shifted by one event, for grpoverlap=0 adjacent groups are independent and share no events.

(startenergy = -1) [real]
Beginning of energy range in eV over which the spectra are collected. If startenergy is negative, the first energy is automatically determined by the smallest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(stopenergy = -1) [real]

End of energy range in eV over which the spectra are collected. If stopenergy is negative, the final energy is automatically determined by the largest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(extraspread = 500) [real]
Energy in eV by which the energy range is extended on either side beyond the smallest and largest energy in the linefitfile for the selected calibration feature. This parameter may be overridden by the startenergy and stopenergy parameters.

(evchannel = 100) [real]
Conversion factor from channel number (for energycol) to energy [eV/chan].

(binwidth = 1.0) [integer]
Energy bin width, in units of channels, to use when collecting spectra.

(broadening = 200) [real]
FWHM of the Gaussian in eV used to initially broaden the theoretical line profile. If fitwidth is set to no, the profile width is fixed at this value.

(gridprofile = no) [boolean]
If gridprofile is set to yes, only output the theoretical profile including any convolution due to the broadening parameter; no fitting is conducted (yes/[no]).

(fitwidth = yes) [boolean]
If fitwidth is set to yes, then fit the width of each spectra in addition to the energy shift (yes/[no]).

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE).

(spangti = yes) [boolean]
If spangti is set to yes, events in different intervals in gtifile may be collected in the same spectrum to be fit. If spangti is set to no, then groups of events used to construct the spectra must be from the same GTI. This parameter is ignored if gtifile is set to NONE (yes/[no]).

(avgwinrad = -1.) [real]
Radius of interval (in units of binwidth) used only to update the initial shift estimate prior to fitting. If avgwinrad is set to -1, this radius is automatically calculated based on the theoretical line profile and the broadening parameter. This is not used in calculating the average results.

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no]).

(writeerrfunc = no) [boolean]
Output the array of chi-squared and likelihood calculated for the SHIFT and WIDTH (yes/[no]).

(minwidth0 = 1.0) [real]
Smallest width, in units of binwidth, allowed as the initial value in width fitting. This parameter provides a lower limit to the initial estimate of the width as computed by the fitting algorithm. The value must be greater than zero.

(maxitcylce = 5) [integer]
Maximum number of fitting iterations.

(r2tol = .01) [real]
Convergence criterion on R-squared for least-squared fitting. Once R-squared changes by less than this amount between fitting iterations, the procedure is finished. This parameter should not normally need to be changed from the default value.

(searchstepshift = 2.) [real]
Step size, in units of binwidth, used when searching for best-fit energy shift in either direction from the initial shift estimate based on the spectrum average. The final shift is obtained using the bisection method (see bisectolshift).

(maxdshift = 5.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of energy shift. If no solutions are found within this deviation at smaller or larger shifts from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolshift = .1) [real]

When the bisection method determines the energy shift to within this amount in units of binwidth, the fitting procedure is completed.

(searchstepwidth = 5.) [real]
Step size, in units of binwidth, used when searching for best-fit convolution width in either direction from the initial width estimate based on the difference between the profile and spectrum statistical variances. The final width is obtained using the bisection method (see bisectolwidth).

(maxdwidth = 10.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of convolution width. If no solutions are found within this deviation at smaller or larger widths from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolwidth = .2) [real]
When the bisection method determines the convolution width to within this amount in units of binwidth, the fitting is procedure is completed.

(minwidth = .5) [real]
Since the least-squares fitting functional is undefined when the width is zero, one must define a minimum allowed fitted width. If the fitting routine attempts to fit a width smaller than this value (in units of binwidth), the fitting procedure fails for the spectrum.

(nerrshift = 100) [integer]
Number of shift values in uncertainty calculations.

(nerrwidth = 100) [integer]
Number of width values in uncertainty calculations.

(shifterrfac = 3.0) [real]
Factor for determining domain of shift uncertainty arrays.

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Compute the gain correction for an HXI event file using the MnK lines as the theoretical profile. The theoretical profile is convolved with a 200 eV Gaussian and gains are computed separately for each SIDE (the default values).
hxigainfit infile=event_in.fits outevtsuffix=clean outfile=drift_out.fits

---

NAME hxinxbgen -- Create a Non-X-ray Background (NXB) spectrum for HXI

USAGE hxinxbgen infile ehkfile regfile innxbfile innxbehk innxbhk outpifile

DESCRIPTION
'hxinxbgen' generates a PI spectrum of the non-X-ray background (NXB) for an HXI observation. The NXB spectrum is calculated from a calibration file containing night Earth data, when X-rays are blocked from the detector's view, accumulated over a period of time. This spectrum is used in spectral fitting (e.g. with XSPEC) to subtract the effects of the particle background from the science data.

'hxinxbgen' uses the following inputs: (a) the science HXI event file for which the NXB spectrum is calculated ('hxinxbgen' reads and uses information in the header keywords related to the time of the observation, and the pointing, as well as the GTI); (b) the extended housekeeping (EHK) file ('ehkfile') containing information on the spacecraft orbit during the observation; (c) The NXB event file (parameter 'innxbfile'); the NXB EHK file ('innxbehk'); (d) a DS9-format region file ('regfile') describing the spectral extraction HXI source region. The region may be specified in RAW, DET, FOC, or SKY coordinates (see 'regmode' parameter).

'hxinxbgen' first filters the NXB event data on the region file. This selection is done using DET coordinates and the task internally transforms the coordinates if the region file is not provided in DET. The task then creates a GTI to apply to the NXB event and NXB EHK data using the 'timefirst' and 'timelast' parameters. These extend the NXB GTI beyond that of the science data to ensure sufficient statistics in the output NXB spectrum. A baseline default value, based on experience with previous missions, is to have a window of 300 days centered on the observation. 'hxinxbgen' then screens the NXB events using the same criteria used for selection of events in the

science data (see 'expr' parameter). Finally the NXB spectrum is produced from NXB events within this GTI that are selected and weighted based on the geomagnetic Cut-Off Rigidity (COR), an estimate of the shielding provided by the Earth's magnetic field against impinging charged particles, and on the time since the last passage through the South Atlantic Anomaly (SAA), an area of enhanced background. NXB spectra are extracted based on the distributions of COR and T_SAA that are present in the science event data. For each science event data T_SAA bin, 'hxinxbgen' extracts spectra from each COR bin in the science data filtered on that combination of T_SAA and COR. The spectrum is weighted by the ratio of the science exposure time in that combination of T_SAA and COR bin to the total science exposure time. The NXB output spectrum is then the sum of these weighted spectra.

The EHK file contains several Cut-Off Rigidity values. Empirically derived values are stored in the columns COR, COR2, or COR3; and a calculated value in the CORTIME column. The choice of COR to use is specified by the 'sortcol' parameter. COR3 is the recommended table to use for HXI. The 'sortbin' parameter specifies the COR value bin boundaries that are used in the NXB selection. The EHK file also contains several values of the time since SAA passage based on the longitude and latitude of the vertices of polygons that define the SAA for various instruments. The choice of T_SAA to use is specified by the 'tsaacol' parameter. The 'tsaabin' parameter specifies the T_SAA value bin boundaries that are used in the NXB selection. Times with COR below the minimum or above the maximum value in 'sortbin', or T_SAA below the minimum or above the maximum in 'tsaabin', are excluded from the NXB data. Therefore these ranges should match any COR and T_SAA filtering performed on the science data.

'hxinxbgen' outputs the weighted NXB PI spectrum ('outpifile'); and optionally, the EHK file corresponding to the science GTI ('outehkfile'); the calibrated, screened, and time-filtered NXB event list ('outnxbfile'); and the NXB EHK file corresponding to that NXB file ('outnxbehk').

PARAMETERS
infile [filename]
Input event file. Header keywords are read from the EVENTS extension, and the GTI extension is used to construct the weighting histogram.

ehkfile [filename]
Input EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

regfile [filename]
Input region file in DS9 format. The region should be in coordinates specified by regmode.

innxbfile [filename]
Input NXB event file used to construct the NXB spectrum.

innxbehk [filename]
Input NXB EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

outpifile [filename]
Output PI spectrum file name.

(outehkfile = NONE) [filename]
Output EHK file. Contains the EHK data for only the times in the infile GTI. If the parameter is set to NONE this file is not created.

(outnxbfile = NONE) [filename]
Output NXB file. Contains the filtered NXB events used in the output spectrum. If the parameter is set to NONE this file is not created.

(outnxbehk = NONE) [filename]
Output NXB EHK file. Contains the EHK data for only the times covered by the filtered NXB events. If the parameter is set to NONE this file is not created.

(regmode = SKY) [string]
Region mode. Specifies the coordinate system used by regfile. Allowed systems are SKY, DET, FOC, and RAW.

(timefirst = 150) [integer]
Days before the science observation used to extract NXB.

(timelast = 150) [integer]
Days after the science observation used to extract NXB.

(sortcol = COR3) [string]
Column for sorting NXB data by COR. This column must exist in the EHK files, and must be either COR, COR2, COR3, or CORTIME.

(sortbin = 0,4,5,6,7,8,9,10,11,12,13,99) [string]
Bin boundaries for sorting NXB data by COR. Times where sortcol is below the minimum or above the maximum value are excluded. This range should match any COR filtering performed on the science data. List must be comma-separated in increasing numerical order.

(tsaacol = T_SAA_HXI1) [string]
Column for sorting NXB data by time since SAA. This column must exist in the EHK files.

(tsaabin = 500,1000,2000,5000) [string]
Bin boundaries for sorting NXB data Time since SAA [s]. Times where tsaabin is below the minimum or above the maximum value are excluded. This range should match any COR filtering performed on the science data. List must be comma-separated in increasing numerical order.

(expr = NONE) [string]
Additional expression to select good events applied to the NXB event list. This should match the screening of the science data. If the parameter is set to NONE no expression is used.

(cleanup = yes) [boolean]
Delete intermediate files ([yes]/no).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
Run hxinxbgen with a restrictive COR range, to reduce the background for a low-surface-brightness target.
hxinxbgen infile=target.evt ehkfile=ehk.fits regfile=target_sky.reg innxbfile=nxb.fits innxbehk=nxb.ehk outpifile=target_hxinxb.pi \
sortbin="6,7,8,9,10,11,12,13,99"

NAME hxipipeline - HXI reprocessing tool

USAGE hxipipeline indir outdir steminputs entry_stage exit_stage

DESCRIPTION
hxipipeline duplicates most of the pipeline (not trend data) for the hxi. It allows the user to run all or part of the pipeline processing and to vary the calibration files and filtering (screening) criteria used. A number of other pipeline processing parameters can also be changed.

HXI Pipeline Stages. The hxipipeline is divided into 3 stages:
Calibration
Data screening
Product creation

Each stage may be run singly or in combination with the preceding stages. This is controlled by the entry_stage and exit_stage parameters.

HXI Stage 1 consists of the following ordered steps:
Run cams2att*
Run hxisgdsff
Run hxisgdpha
Run hxievtid
Run coordevt

*Note: cams2att us automatically run twice. Once with parameters set for HXI1, the other with parameters set for HXI2.

The data screening (Stage 2) is identical to that in the production pipeline, when default parameters are used. For details on the default screening applied to the HXI events (respectively), see:
ahfilter - Create the EHK from attitude & orbital data and create the MKF from housekeeping data based on the CALDB mkfconf file
ahgtigen - Create the GTI from the EHK and MKF parameters based on CALDB selection file
ahscreen - Screen the data based on GTI and CALDB selection file
Default GTI used for screening data are:

GTIPOINT
GTITEL
GTIEHK
GTIMKF

The cleaning process occurs twice for HXI. Once to produce cleaned event files, the second to produce pseudo event files.

The product creation (Stage 3) is identical to that in the production pipeline, when default parameters are used. For HXI events extractor is run on SKY coordinates and a lightcurve, spectra and images are created for each cleaned event file.

INPUT
The input to hxipipeline is specified using (at minimum) the indir parameter. This should be specified as the hxi event_uf level sequence directory, e.g.:

hxipipeline indir=/path/to/100039010/hxi/event_uf

Paths to specific hxi housekeeping and satellite data files can be specified using the attitude, housekeeping, extended_housekeeping, makefilter, orbit and obsgti parameters. The attitude, orbit and hxi housekeeping files are required for stage 1 calibration.

OUTPUT
Filenames, etc.

The number of output files depends on both pipeline processing stage(s) and the options selected. All output files are written to the directory specified by the outdir parameter. The archive directory structure is NOT reproduced (i.e. all output files are in a single directory).

The names of files produced are the same as those found in the HEASARC archive. However the usual "ahXXXXXXXXX" prefix, where "XXXXXXXXX" is the sequence number, can be replaced by a character string set by the stemoutputs parameter. This defaults to the value set by the steminputs parameter.

PARAMETERS
indir [string]
Input directory

outdir [string]
Output directory

steminputs [string]
stem inputs

stemoutputs [string]
stem outputs

instrument [string]
Instrument (HXI,HXI1,HXI2)
entry_stage [integer]
Entry stage

(exit_stage = 2) [integer]
Exit stage

(hxi_start = 0.0) [real]
HXI CALDB start time

verify_input [boolean]
Verify with ftverify (yes, no)

(hxi_mkflabel = HXISFFA1CAM) [string]
Label to use for HXI MKF GTI creation
For pseudo events "PSE" replace CAM in the label

(hxi_ehklabel = HXISFFA1CAM) [string]

Label to use for HXI EHK GTI creation
For pseudo events "PSE" replace CAM in the label

(hxi_evtlabel = HXISFFA1CAM) [string]
Label to use for HXI event screening
For pseudo events "PSE" replace CAM in the label

(ra = -999.99999) [real]
The Right Ascension in decimal degrees of the point to appear in the center of SKY coordinate images. If ra is within the range 0-360 deg., then the value of the parameter is used. If ra is outside this range, then the ra value is read from the event header of the input file, searching first for the RA_NOM keyword and if found using its value, or if not found, then searching for the RA_PNT keyword and using its value. If neither keyword is found, then coordevt exits with an error. The default value for ra triggers the keyword look up.

(dec = -999.99999) [real]
The declination in decimal degrees of the point to appear in the center of SKY coordinate images. If dec is within the range -90- +90 deg., then the value of the parameter is used. If dec is outside this range, then the dec value is read from the event header of the input file, searching first for the DEC_NOM keyword and if found using its value, or if not found, then searching for the DEC_PNT keyword and using its value. If neither keyword is found, then coordevt exits with an error. The default value for dec triggers the keyword look up.

(roll = 0.0) [real]
The roll angle about the center of the SKY coordinate system in decimal degrees. The roll angle is the angle measured counterclockwise from Celestial North to the positive SKY Y axis.

(hx1_optdetx = -999.99999) [real]
HXI1 optical detx coordinate

(hx1_optdety = -999.99999) [real]
HXI1 optical dety coordinate

(hx1_optfocx = -999.99999) [real]
HXI1 optical focx coordinate

(hx1_optfocy = -999.99999) [real]
HXI1 optical focy coordinate

(hx1_optskyx = -999.99999) [real]
HXI1 optical skyx coordinate

(hx1_optskyy = -999.99999) [real]
HXI1 optical skyy coordinate

(hx2_optdetx = -999.99999) [real]
HXI2 optical detx coordinate

(hx2_optdety = -999.99999) [real]
HXI2 optical dety coordinate

(hx2_optfocx = -999.99999) [real]
HXI2 optical focx coordinate

(hx2_optfocy = -999.99999) [real]
HXI2 optical focy coordinate

(hx2_optskyx = -999.99999) [real]
HXI2 optical skyx coordinate

(hx2_optskyy = -999.99999) [real]
HXI2 optical skyy coordinate

(hx1_ra_pnt = -999.99999) [real]
RA of HXI1 pointing [deg]

(hx1_dec_pnt = -999.99999) [real]
DEC of HXI1 pointing [deg]

(hx2_ra_pnt = -999.99999) [real]
RA of HXI2 pointing [deg]

(hx2_dec_pnt = -999.99999) [real]
DEC of HXI2 pointing [deg]

attitude [string]
hxipipeline:
The name of the attitude file used in the conversion to SKY coordinates. If the transformation does not involve SKY coordinates, then an attitude file need not be provided. The attitude can be provided as either quaternions or Z-Y-Z Euler angles. The attitude format in the attfile must match the value of the attform parameter.

orbit [string]
hxipipeline:
The name of a FITS satellite orbit file. This file provides the satellite orbital velocity as a function of time, which is used in calculating the orbital aberration correction. There are two orbit formats supported (VECTOR, e.g. Swift, and COMPONENT, e.g. Suzaku). The orbit format in the orbitfile must match the value of the orbform parameter. If the orbital aberration correction is not required (i.e. orbaberration=no), then an orbit file need not be provided.

(extended_housekeeping = ah1001.ehk) [string]
ahgtigen:
Extended housekeeping file

(makefilter = ah1001.mkf) [string]
ahgtigen:
Makefilter file

(obsgti = ah1001_gen.gti) [string]
ahscreen:
Observation GTI file

(regionfile = NONE) [string]
extractor:
Input region file

(hx1teldef = CALDB) [filename]
coordevt/cams2att:
Name of HXI1 file. If hx1teldef=CALDB, then the TelDef file for the instrument specified by detnam is used to determine the appropriate TelDef file from a CALDB query.

(hx2teldef = CALDB) [filename]
coordevt/cams2att:
Name of HXI2 file. If hx2teldef=CALDB, then the TelDef file for the instrument specified by detnam is used to determine the appropriate TelDef file from a CALDB query.

(remapfile = CALDB) [filename]
hxisgdsff/hxievtid:
File describing the remapping of ASIC_ID and READOUT_ID to sequential numbers in the columns ASIC_ID_RMAP and READOUT_ID_RMAP. Specify CALDB to retrieve the file from the calibration database.

(gainfile = CALDB) [filename]
hxisgdpha:
FITS table describing the gain calibration of PHA for the HXI or SGD camera. The domain of PHA values is divided into intervals, and a set of cubic polynomial coefficients is given for each interval. Specifying CALDB causes this file to be retrieved from the calibration database.

(badpixfile = CALDB) [filename]
hxisgdpha/hxievtid:

FITS table giving the active and bad channels (pixels) of the HXI or SGD camera. Specifying CALDB causes this file to be retrieved from the calibration database.

(fluorefile = CALDB) [filename]
hxievtid:
Calibration file containing the energy ranges of fluorescence electrons in CdTe and Si. Default: CALDB.

(enecutfile = CALDB) [filename]
hxievtid:
Input energy cut file

(cm1teldef = CALDB) [filename]
cams2att:
Name of CAMS1 TelDef file. If cams1teldef=CALDB, then the telescope and instrument specified by keywords in infile1 are used to determine the appropriate TelDef file from a CALDB query.

(cm2teldef = CALDB) [filename]
cams2att:
Name of CAMS2 TelDef file. If cams2teldef=CALDB, then the telescope and instrument specified by keywords in infile1 are used to determine the appropriate TelDef file from a CALDB query.

(camstempxy = CALDB) [filename]
cams2att:
Name of the CAMS temperature compensation file. If camstempxy=CALDB, then the telescope and instrument specified by keywords in infile1 are used to determine the appropriate calibration file from a CALDB query.

(leapsecfile = REFDATA) [filename]
ahgtigen/ahscreen:
Input leap second file (or CALDB, [REFDATA])

(selectfile = CALDB) [filename]
ahgtigen/ahscreen:
Input file with the selection expressions

(outnsubcol = no) [boolean]
hxisgdpha:
Output the PHA_NSUB column (yes/no). This column is the PHA minus the ASIC common mode noise (ASIC_CMN column in the SFF).

(datamode = NONE) [string]
hxisgdpha:
Substitute DATAMODE in place of event value (or NONE)

(randomize = yes) [boolean]
hxisgdpha:
Allow randomization ([yes]/no)

(rejectbgo = no) [boolean]
hxievtid:
Reject events for which BGO trigger occurred (yes/[no]).

(skipreco = no) [boolean]
hxievtid:
If yes, READALL and CALMODE occurrences is not reconstructed. Default: NO.

(outcalfile = NONE) [filename]
hxievtid:
Output a file containing the reconstruction information for each occurrence by detector side. This file can be used as input to the gainfit tool.

(startsys = RAW) [string]
cams2att:

Origin coordinate system for transformation done by the tool ahdet2att.

(startstep = 1) [integer]
cams2att:
The beginning step for processing within this tool (1-5). The five processing steps are documented in the help file for cams2att.

(stopstep = 5) [integer]
cams2att:
The final step for processing within this tool (1-5). The five processing steps are documented in the help file for cams2att.

(inext = EVENTS) [string]
cams2att:
Name of data extension in infile1 and infile2.

(outext = CAMS_OFFSETS) [string]
cams2att:
Name of data extension in the output file from step 1 (see tempcorfile1 and tempcorfile2).

(flipsign = no) [boolean]
cams2att:
Flip sign of output offsets and angles (yes/[no])

(prefiltfile1 = NONE) [string]
cams2att:
Name of filtered, temperature-corrected CAMS1 displacement file. This file is written by the tool ftselect in Step 2, and it is read by the tool cams2det in Step 3. If startstep=1 or 2, then specifying "NONE" causes the file to be given the name tmp_prefiltfile1.fits and to be deleted if cleanup=yes. If startstep=3, then the value "NONE" is invalid, and an actual filename must be specified.

(prefiltfile2 = NONE) [string]
cams2att:
Name of filtered, temperature-corrected CAMS2 displacement file. This file is written by the tool ftselect in Step 2, and it is read by the tool cams2det in Step 3. If startstep=1 or 2, then specifying "NONE" causes the file to be given the name tmp_prefiltfile2.fits and to be deleted if cleanup=yes. If startstep=3, then the value "NONE" is invalid, and an actual filename must be specified.

(filtoffset = NONE) [string]
cams2att:
Name of intermediate offset file after filtering. This file is written by the tool ftselect in step 4 and read by the tool ahdet2att in Step 5. If startstep=1, 2, 3, or 4, then specifying "NONE" causes the file to be given the name tmp_filtoffset.fits and to be deleted if cleanup=yes. If startstep=5, then the value "NONE" is invalid, and an actual filename must be specified.

(prefiltexpr = DSP_UP==1 && IS_SAMPLING==1) [string]
cams2att:
Expression used by ftselect to filter the temperature corrected CAMS data files tempcorfile1 and tempcorfile2.

(filtexpr = BAD_UNITS==0) [string]
cams2att:
Expression used by ftselect to filter offsetfile.

(gtiexpr0 = BAD_UNITS==0) [string]
cams2att:
Expression used to create GTI for both CAMS units.

(gtiexpr1 = BAD_UNITS==2) [string]
cams2att:
Expression to create GTI for CAMS1

(gtiexpr2 = BAD_UNITS==1) [string]
cams2att:
Expression to create GTI for CAMS2

(deltaxcol = DELTARAWX) [string]
cams2att:

Column name for change in X coordinate in the file filtoffset read by Step 5.

(deltaycol = DELTARAWY) [string]
cams2att:
Column name for change in Y coordinate in the file filtoffset read by Step 5.

(sincol = SINANGLE) [string]
cams2att:
Column name for sine of rotation angle in the file filtoffset read by Step 5.

(coscol = COSANGLE) [string]
cams2att:
Column name for cosine of rotation angle in the file filtoffset read by Step 5.

(coordevt_startsys = LOWEST) [string]
coordevt:
Sets starting coordinate system specifically for coordevt. The name of the coordinate system from which the coordinate conversions begin. If startsys=LOWEST, then the coordinate transformation begins with the lowest level system in the teldef file (COORD0; usually RAW). The startsys and stopsys parameters work together to specify the origin (startsys) and final (stopsys) systems. Setting startsys=LOWEST and stopsys=HIGHEST the task converts the bottom-level coordinates into all the other coordinate systems. It is allowed for startsys to be a higher level coordinate system than stopsys. For example, setting startsys=HIGHEST and stopsys=LOWEST converts the top-level coordinates into all the other coordinate systems.

(stopsys = HIGHEST) [string]
coordevt:
The name of the coordinate system at which the coordinate conversions end. If stopsys=HIGHEST, then the coordinate transformation chain ends with the highest level system (SKY). See also the description of the startsys parameter.

(annaber = no) [string]
coordevt:
If YES, the effects of annual aberration are taken into account when calculating the SKY coordinate values. Annual aberration is the apparent bending of light due to the Earth's orbit around the Sun. This is at most a ~20.49 arcsec effect. If INVERT, the sign of the annual aberration correction is flipped before being applied to the sky coordinates. This is used for debugging. If NO, the annual aberration is not corrected.

(followsun = no) [boolean]
coordevt:
Should aberration be recalculated for each event time? If set to "no" the aberration at the time given by the MJDOBS keyword is used for all events. Setting this to "no" is acceptable except for very long observations, but makes the program run slightly faster. The "yes" setting should be used for highest accuracy, and this setting uses the MJDREF keyword (or pair of MJDREFI and MJFREFF keywords) along with the TIME column of the event extension to calculate absolute event times.

(orbaber = no) [string]
coordevt:
If YES, the effects of orbital aberration are taken into account when calculating the SKY coordinate values. Orbital aberration is the apparent bending of light due to the satellite's orbit around the Earth. For a satellite in low-earth orbit, this is at most a ~5 arcsec effect. If INVERT, the sign of the orbital aberration correction is flipped before being applied to the sky coordinates. This is used for debugging. If NO, the orbital aberration is not corrected.

(attinterp = LINEAR) [string]
coordevt:
Spacecraft attitude interpolation method. The value LINEAR results in the attitude being linearly interpolated to the event time. The value CONSTANT results in the attitude being taken from the nearest attitude record.

(dattinterp = LINEAR) [string]
coordevt:
Delta attitude interpolation method. The value LINEAR results in the delta attitude being linearly interpolated to the event time. The value CONSTANT results in the delta attitude being taken from the nearest attitude record.

(attdt = 32.) [real]
coordevt:

Allowed time in seconds to extrapolate the spacecraft attitude before the first or after the last row in the attitude file. Events beyond this time margin have their coordinates set to a null value.

(dattdt = 0.5) [real]
coordevt:
Allowed time in seconds to extrapolate the auxiliary ("delta") attitude before the first or after the last row in the attitude file. Events beyond this time margin have their coordinates set to a null value.

(chkattgap = no) [boolean]
coordevt:
Set this parameter to "yes" to use the attdt also for interpolation. Event times that fall within the sky attitude time range but not within the attdt of the nearest attitude time is set to null. Set chkattgap to "no" to always interpolate the attitude at the event time. The attdt value is used for extrapolation regardless of the value of chkattgap.

(chkdattgap = yes) [boolean]
coordevt:
Set this parameter to "yes" to use the dattdt also for interpolation. Event times that fall within the delta attitude time range but not within the dattdt of the nearest attitude time is set to null. Set chkdattgap to "no" to always interpolate the attitude at the event time. The dattdt value is used for extrapolation regardless of the value of chkdattgap.

(attext = ATTITUDE) [string]
coordevt:
The name of the FITS attitude file extension containing satellite attitude data.

(attcol = QPARAM) [string]
coordevt:
The name of the FITS column containing attitude values as a vector in the attext extension of the attitude table.

(attform = QUAT) [string]
coordevt:
The format in which the attitude is provided in the attitude table. Two formats are supported. The QUAT format is a quaternion [x, y, z, real]. The EULER format is a Z-Y-Z Euler angle trio [phi, theta, psi] in degrees.

(orbext = ORBIT) [string]
coordevt:
The name of the FITS orbit file extension containing satellite orbit data.

(orbcol = VELOCITY) [string]
coordevt:
The name(s) of the FITS column(s) containing orbit values in the orbext extension of the orbit file. If orbform="VECTOR", then orbcol is the name of the single FITS column containing the orbit values as a vector. If orbform="COMPONENTS", then orbcol must be a string containing three comma-separated column names, specifying in order the X, Y and Z components of the orbital velocity.

(orbform = VECTOR) [string]
coordevt:
The format in which the orbital velocity is provided in the orbit file. Two formats are supported. For the VECTOR format, the velocity is provided as a vector column with three elements (X, Y and Z in Earth-Centered Inertial (ECI) system). For the COMPONENT format, the velocity is provided in three separate columns.

(coordevt_randomize = TELDEF) [string]
coordevt:
Set randomize coordinates specifically for coordevt. If this parameter is set to "no", coordevt assumes that each event occurred at the center of its coordinate pixel. If it is set to "yes", the coordinates from system randsys onward is calculated assuming a random location within the randsys system pixel. If randomize="CALDB", then randomization is enabled (RANCOORD != 'NONE') or disabled (RANCOORD = 'NONE') depending on the TelDef keyword RANCOORD. This parameter only controls randomization in transformations previous to the transformation to the SKY system.

(randsys = TELDEF) [string]
coordevt:
The name of the starting coordinate system for which randomization is performed, as long as randomization is enabled by the randomize parameter. If randsys="CALDB", the value of the RANCOORD keyword (if present) of the TelDef file specifies this system.

(randscalesys = TELDEF) [string]
coordevt:
The name of the coordinate system whose pixels determine the size of the randomization if randomization is enabled by the randomize parameter. If randscalesys="CALDB", the value of the RAN_SCAL keyword of the TelDef file is used if the keyword is present, and the value of the randsys parameter is used if the RAN_SCAL keyword is not present. It is most common for randsys and randscalsys to be the same coordinate system, but it is not required.

(infileext = EVENTS) [string]
coordevt:
The name of the infile extension containing the event table.

(timecol = TIME) [string]
coordevt:
The name of the FITS column in the event table containing time values.

(inclfloatcol = no) [boolean]
coordevt:
If this parameter is set to "yes", an additional column of unrounded coordinate values is included in the output event file for each coordinate of each system after the startsys system. Enabling this parameter is useful for stringing multiple coordevt runs together (e.g., convert RAW to DET in one run and DET to SKY in a second run) without loss of precision due to the normal rounding of coordinates. Use startwithfloat="yes" for runs after the first so that the unrounded coordinates from the previous coordevt run are used as the starting coordinate values for a subsequent run. Rounded coordinate columns are written regardless of the value of this parameter. Unrounded SKY coordinate columns are included if either of inclfloatcol or inclfloatskycol is set to "yes".

(inclfloatskycol = no) [boolean]
coordevt:
If this parameter is set to "yes", an additional column of
unrounded coordinate values is included in the output event file
for each SKY coordinate. Rounded SKY coordinate columns are
included regardless of the value of this parameter. Unrounded
SKY coordinate columns are included if either of inclfloatcol or
inclfloatskycol is set to "yes".

(floatcolsuffix = _FLOAT) [string]
coordevt:
This parameter sets the suffix used in all unrounded coordinate column names. If the rounded coordinate column name is DETX and floatcolsuffix=_FLOAT, then the corresponding unrounded coordinate column name is DETX_FLOAT. This parameter is used only when either of inclfloatcol or inclfloatskycol is set to "yes".

(startwithfloat = no) [boolean]
coordevt:
This parameter is set to "yes" to use the unrounded coordinate column values (e.g., floating point values from DETX_FLOAT and DETY_FLOAT columns) as the starting coordinate columns. Enabling this parameter is useful for non-initial runs of coordevt in a string of runs that each convert between a subset of the transformations. The inclfloatcol parameter must be set to "yes" for startwithfloat to have an effect. When either of inclfloatcol or startwithfloat is disabled, rounded coordinate values (e.g., integer values from DETX and DETY columns) are used as the starting coordinate values.

(blankcol = yes) [boolean]
coordevt:
This parameter is set to "yes" to fill the coordinate columns after the stopsys coordinate system with NULL values. If blankcol is set to "no", such columns is not changed. The behavior is somewhat different depending on what coordinate columns already exist in the input file. If columns for coordinates beyond the stopsys coordinate system exist in the input file, then when blankcol="yes", their values are replaced with NULL values and when blankcol="no", their values are copied to the output file unchanged. If, however, such columns do not already exist in the input file, then they are created and filled with NULL values when blankcol="yes", but not created if blankcol="no".

(btnull1 = 255) [integer]
coordevt:
This parameter sets the null integer value for any output unsigned byte (TFORM = "1B") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(itnull1 = -999) [integer]
coordevt:
This parameter sets the null integer value for any output short integer (TFORM = "1I") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(jtnull1 = -999) [integer]
coordevt:
This parameter sets the null integer value for any output long integer (TFORM = "1J") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(ktnull1 = -999) [integer]
coordevt:
This parameter sets the null integer value for any output long integer (TFORM = "1K") coordinate columns that are missing or lack the TNULL keyword in the input event file.

(sbtnull1 = 255) [integer]
coordevt:
This parameter sets the null integer value for any output signed byte (TFORM = "1B" with TZERO = -128) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the signed domain.

(uitnull1 = -999) [integer]
coordevt:
This parameter sets the null integer value for any output unsigned short integer (TFORM = "1I" with TZERO = 32768) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the unsigned domain.

(ujtnull1 = -999) [integer]
coordevt:
This parameter sets the null integer value for any output unsigned long integer (TFORM = "1J" with TZERO = 2147483648) coordinate columns that are missing or lack the TNULL keyword in the input event file. The specified null value is the value to be stored in the FITS file, not the value in the unsigned domain.

(occurrenceid = -1) [integer]
hxievtid:
If greater than zero, only expand this single occurrence.

(seed = 0) [integer]
coordevt/cams2att:
Random number generator seed to be used if randomization is enabled. For a given seed value, the same results is produced each time the program is run.

(stemreport =) [string]
File stem for log and temporary files. If the parameter is not set the script automatically sets the stem to "hxipipeline_YYYYMMDDTHHMMSS_" and appends log file and temp file names as needed. Intended to be set by ahpipeline.

(numerrs = 0) [string]
Number of errors from hxipipeline (output)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

These parameters are used to specify default null value processing. Null values may appear for several reasons, including the following: 1. the event time cannot be matched to a row of an attitude file within the time margin; 2. the coordinate system is subsequent to the stopsys coordinate system and blankcol is enabled; 3. a calculated coordinate is outside the allowed range of that coordinate. If an output column already exists in the input file and TNULL for that column is specified in the event file header, then the existing TNULL keyword is copied to the output column. If, however, the output column does not exist in the input file or the column exists, but does not have an associated TNULL keyword, then the appropriate nullvalue parameter is used.

THE FOLLOWING PARAMETERS ARE SET BY THE SCRIPT INTERNALLY:

CAMS2ATT:
infile1 = cams1_data.fits [filename]
HXIPIPELINE looks for a file with the appropriate nomenclature.
If found, this parameter is set to that file. Otherwise it is set to "NONE" It is possible for either the cm1 or cm2 data file to be set to NONE, however the script fails if both infile1 and infile2 are missing. Name of input FITS file containing the X and Y positions determined by the CAMS1 system. The tool does not change this file.

infile2 = cams2_data.fits [filename]
HXIPIPELINE looks for a file with the appropriate nomenclature.
If found, this parameter is set to that file. Otherwise it is set to "NONE" It is possible for either the cm1 or cm2 data file to be set to NONE, however the script fails if both infile1 and infile2 are missing. Name of input FITS file containing the X and Y positions determined by the CAMS2 system. The tool does not change this file.

outfile = deltaq.fits [filename]
Set to ${outdir}/${stemoutputs}hx1.att the first time cams2att runs.
Set to ${outdir}/${stemoutputs}hx2.att the second time cams2att runs.
Name of output file containing the time-dependent delta-attitude quaternions for conversion from the HXI RAW to ACT system. The tool creates this file.

(tempcorfile1 = NONE) [string]
Set to the same name as infile1, but in the outdir directory the first time cams2att runs.
Set to "NONE" the second time cams2att runs."
Name of temperature-corrected CAMS1 displacement file. This file is written by the tool ftcalc in Step 1, and it is read by the tool ftselect in Step 2. If startstep=1, then specifying "NONE" causes the file to be given the name tmp_tempcorfile1.fits and to be deleted if cleanup=yes. If startstep=2, then the value "NONE" is invalid, and an actual filename must be specified.

(tempcorfile2 = NONE) [string]
Set to the same name as infile2, but in the outdir directory the first time cams2att runs.
Set to "NONE" the second time cams2att runs."
Name of temperature-corrected CAMS2 displacement file. This file is written by the tool ftcalc in Step 1, and it is read by the tool ftselect in Step 2. If startstep=1, then specifying "NONE" causes the file to be given the name tmp_tempcorfile2.fits and to be deleted if cleanup=yes. If startstep=2, then the value "NONE" is invalid, and an actual filename must be specified.

(offsetfile = NONE) [string]
Set to ${outdir}/${stemoutputs}hx1_cms.fits the first time cams2att runs.
Set to ${outdir}/${stemoutputs}hx2_cms.fits the second time cams2att runs.
Name of intermediate offset file. This file is written by the tool cams2det in step 3 and read by the tool ftselect in Step 4. If startstep=1, 2, or 3, then specifying "NONE" causes the file to be given the name tmp_offsetfile.fits and to be deleted if cleanup=yes. If startstep=4, then the value "NONE" is invalid, and an actual filename must be specified.

(hxiteldef = CALDB) [string]
Set to hx1teldef parameter the first time cams2att runs.
Set to hx2teldef parameter the second time cams2att runs.

detnam [string HXI1, HXI2]
Set to HXI1 the first time cams2att runs.
Set to HXI2 the second time cams2att runs.
Name of HXI unit for which output is calculated

HXISGDSFF:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization.
Input First FITS File (FFF) from HXI or SGD.

outfile [filename]
Set to the same name as infile, but in the outdir directory.
Output Second FITS File (SFF). May be the same or a different file from infile. If outfile is the same file as infile, then clobbering must be enabled or the tool exits with an error. Clobbering is enabled by specifying clobber=yes (see below) or by prepending an exclamation point (!) to the output file name. Either of these methods cause the file conversion to be done in-place by modifying the input file.

HXISGDPHA:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff.
Input Second FITS File (SFF) from HXI or SGD.

outfile [filename]
Set to the same name as infile, but in the outdir directory.
Output Second FITS File (SFF). If it is the same as infile, then specifying clobber=yes (see below) causes the file conversion to be done in-place by modifying the input file. Prepending an exclamation point (!) to the output file name has the same effect.

HXISGDEXPAND:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff and hxisgdpha.
Input file of event data from an HXI instrument. This is the Second FITS File (SFF) after PHA calibration by the tool hxisgdpha.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "_uf.evt" to "exp_ufa.evt". Output event file containing results of event reconstruction.

HXIEVTID:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff and hxisgdpha. Input file of event data from an HXI instrument. This is the Second FITS File (SFF) after PHA calibration by the tool hxisgdpha.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "_uf.evt" to "rec_ufa.evt".
Output event file containing results of event reconstruction.

COORDEVT:
infile [filename]:
Set systematically to the elements in an array which is constructed from any unfiltered event files in the input directory. File name and optional extension name or number enclosed in square brackets of the input table that includes the coordinates (usually RAW) to start the first coordinate conversion. The infile is not changed unless the same file is given as the output file.

outfile [filename]:
Name of the output file that contains the calculated values of the transformed coordinates. The outfile is created by copying the infile to outfile, updating the values in the coordinate columns of the event extension, and updating relevant header keywords. Other contents of the infile are preserved in the outfile.

teldeffile [filename]:
The name of the Telescope Definition (TelDef) calibration data base (CALDB) file, which specifies the coordinate systems and transformation properties. When this parameter is set to "CALDB", a TelDef file for the telescope and instrument specified by the TELESCOP and INSTRUME keywords, respectively, is used.

(attfile = NONE) [filename]:
Set to the hxipipeline attitude parameter (see above)

(dattfile = NONE) [string]:
The script reads the instrument name from the header of the uf event file it is processing and uses the appropriate hx1.att or hx2.att file produced by cams2att. The name of or list of the names of the auxiliary ("delta") attitude files needed for certain coordinate transformations (e.g., Hitomi HXI RAW-->ACT). If a list of files is needed, they can be entered one per line in a text file, and then dattfiles should be set to that text filename prefixed with "@", for example dattfiles = "@dattfiles.txt". A delta-attitude file must be provided for each coordinate transformation of type 'SKYATT' in the teldef file, with the exception of the transformation to the SKY system. The delta-attitude must be in quaternion format.

(orbfile = NONE) [filename]
Set to the hxipipeline orbit parameter (see above)

AHGTIGEN:
mkffile [filename]
Set to the makefiler file parameter the first time ahgtigen runs.
Set to the extended_housekeeping file parameter the second time ahgtigen runs (unless the files being generated are pseudo event files).

outfile [filename]
Set to ${hxifile_out} . gtimkf, when the input file is set to the makefiler file parameter.
Set to ${hxifile_out} . gtiehk, when the input file is set to the extended_housekeeping file parameter.

(instrume = NONE) [string]
(HXI1 or HXI2) Read from the header of the file being processed.

(label = NONE) [string]
HXISFFA1CAM or HXISFFA1PSE depending on if the files being generated are "clean" or "pseudo" respectively.

AHSCREEN:
infile [string]
Set systematically to the elements in an array which is constructed from any reconstructed unfiltered event files produced in the calibration stage. If the calibration stage is being skipped, this array is filled from searching the input directory instead.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "rec_ufa.evt" to "rec_cl.evt".

(gtifile) [string]
Set as an array of gtifiles with corresponding extensions; pointing, tel, mkf, and (where applicable) ehk.

(upkeyword = YES) [boolean]
Update timing keywords from input file(s) in output file. Hardcoded as yes

(label = NONE) [string]
HXISFFA1CAM or HXISFFA1PSE depending on if the files being generated are "clean" or "pseudo" respectively.

EXAMPLES
1. The following command recalibrates (stage 1) and re-screen (stage 2) all HXI data for sequence 100039010 that currently resides in the directory /data/100039010/hxi/event_uf, and the output is stored in a directory called /data/100039010_reproc/:

```
 hxipipeline indir=/data/100039010/hxi/event_uf outdir=/data/100039010_reproc entry_stage=1 exit_stage=2 steminputs=ah100039010 \
attitude=/data/100039010/auxil/ah100039010/ah100039010.att orbit=/data/100039010/auxil/ah100039010/ah100039010.orb \
obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk
```

2. The following command re-screens (stage 2 only) HXI data for the same data set as in the previous example.

```
hxipipeline indir=/data/100039010/hxi/event_uf outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 steminputs=ah100039010 \
obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk
```

3. The following command creates products (stage 3 only) HXI data for a calibrated data set:

```
hxipipeline indir=/data/100039010/hxi/event_cl outdir=/data/100039010_reproc entry_stage=3 exit_stage=3 steminputs=ah100039010 \
regionfile=none
```

NOTES
None, but see help for individual parameters above.

NAME hxirspeffimg - Creates a combined effective are and spectral response matrix ("RSP" file) and/or a flat-field correction image, for the HXI1 + HXT1 or HXI2 + HXT2 combinations, accounting for the telescope effective area, detector efficiencies, and CAMS motion.

USAGE hxirspeffimg telescop instrume erange dattfile filtoffsetfile emapfile qefile rmffile onaxisffile onaxiscfile vigfile outflatfile outfile regionfile xrtevtfile sampling

DESCRIPTION

'hxirspeffimg' creates, for the HXI1 + HXT1 or HXI2 + HXT2 combinations:

- A combined effective area and spectral response file (or RSP file) and/or
- A flat-field correction image.

'hxirspeffimg' takes as input an exposure map file, a set of region files in detector coordinates, and an event file containing raytracing events for simulated photons injected into the telescope. The exposure map file is generated by the tool 'ahexpmap' and contains histograms based on discrete satellite attitude pointings. The region files are made by the script 'aharfgen' which uses the attitude information in the exposure map file and the information in a region file in R.A./DEC coordinates to generate several region files in the detector coordinate system, one region file per attitude bin in the exposure map file. The raytracing event file is also made by the script 'aharfgen', which runs the raytracing code 'xrtraytrace' based on information in the attitude histograms in the exposure map file. The event file from previous runs of the raytracing may be used if the arf is recalculting using a different binning (the binning is defined by rmf), different region and whether or not the auxiliary transmission file is used.

The RSP file and flat-field corrections include the effects of the telescope effective area, detector quantum efficiency, satellite attitude variations, and the CAMS motion of the optical bench with respect to the telescope. Note that, since 'hxirspeffing' must consider satellite attitude variations for many CAMS motion time intervals, the run time of 'hxirspeffimg' can be potentially impractical. The parameter 'sampling' allows a trade-off between run time and the accuracy of the final RSP and/or flat-field file, by not considering every time bin in the CAMS input files (see description for parameter 'sampling').

PARAMETERS

telescop [string]
Mission name (value to write in header keyword TELESCOP for CALDB).

instrume [string]
Instrument for which the response matrix (RSP) file is to be made: HXI1 or HXI2.

erange [string]
A string containing two numbers that correspond to the lower and upper energy (in keV) of the bandpass for calculating the flat-field correction image.

dattfile [filename]
Name of the file containing the RAWX, RAWY coordinates of the optical axis as a function of time (the optical axis RAWX, RAWY position moves due to the CAMS motion). The file is made by running cams2att.

filtoffsetfile [filename]
Name of the delta-attitude file describing the CAMS motion with columns DELTARAWX, DELTARAWY, SINANGLE, COSANGLE, as a function of time. The file is made by running cams2att with 'filtoffset=filtoffsetfile'.

emapfile [filename]
Name of the exposure map file (made by the tool 'ahexpmap'), that contains histograms of satellite attitude and related quantities, as well as "good time intervals" (GTI) for the attitude bins.

onaxisffile [filename]
Name of a file and extension that contains the on-axis telescope effective area (appropriate for instrument), pre-calculated on a fine energy grid. This file must also have, in the primary extension, an image that allows 'hxirspeffimg' to correct for the shadowing effects of HXI baffle structure in the flat-field calculations. If the parameter is set to CALDB, the file is read from the calibration database.

onaxiscfile [filename]
Name of a file and extension that contains the on-axis telescope effective area (appropriate for instrument), pre-calculated on a coarse energy grid. If the parameter is set to CALDB, the file is read from the calibration database.

regionfile [filename]
Names of standard SAO ascii region files, or the name of a file containing a list of names of such files. In the latter case, the format is "@aharfgen_regions.lis", where aharfgen_regions.lis is an ascii file in which each row contains the name of a region file. The coordinate system of these region files are detector coordinates, and the region files specify the data extraction regions that correspond to the discrete satellite attitude intervals in the exposure map file ('emapfile') attitude histogram. The region files are only needed for creating the RSP file (response matrix), and not the flat-field file.

xrtevtfile [filename]
Name of the raytracing event file that was created by 'aharfgen'.

outflatfile [filename]
Name of the output image file containing the flat-field efficiency image. If outflatfile is set to NONE, the flat-field calculations are not performed, and no flat-field file is made.

outmaptype [string]
Type of map output. The possibke type are EXPOSURE or EFFICIENCY. The first contains just the time in each pixel given in seconds. The second contains the quantum efficiency and exposure effect normalized by the total therefore each oixel value ranges betaeen 0 and 1. The name of the map is given in outflatfile. The EXPOSURE and/or EFFICIENCY may be written out when the task is either run to create an rsp or a flat field. Only one map may be created for each run.

outfile [filename]
Filename stem for the output response matrix (RSP) file and the output ancillary response file (ARF). The actual filenames are formed by concatenating the stem with extensions '.rsp' and '.arf', respectively. If outfile is set to NONE, the response matrix calculations are not performed, and the RSP file and ARF are not written.

sampling [integer]
Sampling factor for the HXI CAMS delta-attitude bins. 'sampling = 1" uses the original time binning in the input CAMS files ('dattfile' and 'filtoffsetfile'). 'sampling=N' uses the first time bin, and then the (N+1)th bin etc.

(rmfthresh = 1.0e-10) [real]
Define the lower threshold for writing non-zero matrix elements in the HXI output RSP file. If the matrix elements value is less than the value of rmfthresh then that matrix element is et to 0. It is not recommended to increase the value rmfthresh above the default value as it may reduce the compressibility of the output, making the matrix too large for the software to handle.

qefile [filename]
Name of the file containing the quantum efficiency (QE) for the detector, a function of energy for each pixel (i.e. a 128x128 image of QE versus energy functions for each of 5 layers). If the parameter is set to CALDB, the file is read from the calibration database.

rmffile [filename]
Name of the line-spread-function (LSF) file that contains five response matrices (RMFs). One matrix is required for each of the 5 layers of one HXI detector. If the parameter is set to CALDB, the file is read from the calibration database. The energy grid 'rmffile' is the one that is used for final output response matrix (RSP) file. A single output RSP file is created that contains the correct weighting for the responses separated by layer, and includes the telescope effective area, as well as the effects of the CAMS motion. 'rmffile' is not needed for creating the flat-field file.

vigfile [filename]
Name of the calibration file containing coefficients that are used in analytic functions to calculate the telescope vignetting functions, used in the flat-field calculations. If the parameter is set to CALDB, the file is read from the calibration database. 'vigile' is not needed for creating an RSP file (response matrix).

auxtransfile [filename]
Name of the input auxiliary transmission file. This file is used to apply additional transmission that is not accounted in the telescope calibration files used by the raytracing. By default is set to NONE. To apply the auxiliary transmission users has either to input a file or set to CALDB.

(stopsys = SKY) [string]
Name of the coordinate system for the flatfield output. By default is SKY pixels. Other coordintes are ACT, DET or FOC.

(minphoton = 100) [integer]
The minimum number of photons that successfully reach the focal-plane, per raytracing energy grid point, that is acceptable to make a viable response matrix (RSP) file. The number of focal-plane raytraced photons that contribute to the effective area of the telescope must exceed minphoton for every energy, otherwise the program aborts.

(baffle = "64.0 64.0 25.35 622.0 24.67 124.0") [string]
A string list of six numbers that specify parameters of the HXI baffle model as follows: 1st value = Baffle center RAWX coordinate 2nd value = Baffle center RAWY coordinate 3rd value = Radius of top entrance of baffle (mm) 4th value = Height of baffle entrance hole above focal plane (mm) 5th value = Radius of bottom exit of baffle (mm) 6th value = Height of baffle exit hole above focal plane (mm)

(polydeg = "DEFAULT") [string]
Polydeg defines the polynomial order for the fitting. The allowed values are: DEFAULT and 1 to 5 for the HXI and 1 to 10 for the SXS and SXI. For the HXI the DEFAULT value of polydeg is set to 5. For the SXI and SXS, the DEFAULT value is set to the order tested internally for fitting stability.

(buffer = -1) [integer]

Rows to buffer (-1=auto, 0=none, >0=numrows)

[ccldhm]

EXAMPLES

1. Make a response matrix (RSP file), ARF, and flat field for a source in HXI1, given an exposure map file src_expmap.fits made by the tool 'ahexpmap', and a set of region files in detector coordinates that are derived from src_sky.reg using the exposure map src_expmap.fits and the tool 'aharfgen', which should be run prior to running 'hxiespeffimg'. Each region file in detector coordinates corresponds to an attitude histogram bin in the exposure map file. In this example, the names of the region files in detector coordinates are listed, one file per row, in the ascii file aharfgen_region.lis. The raytracing event file that is required by the tool 'hxirspeffimg' is made by the prior run of 'aharfgen'. Note that 'hxirspeffimg' does not need any explicit information about the spatial distribution of the X-ray source because that is already factored into the raytracing event file. In this example, the energy range selected for the flat field image is 8 to 25 keV. The two CAMS files ('dattfile' and 'filtoffsetfile') are made by running 'cams2att' (with 'filtoffset=filtoffsetfile'), prior to running 'hxirspeffimg'.

```
hxirspeffimg telescop=HITOMI
        instrume=HXI1
           erange="8.0 25.0"
           dattfile=hxi1_cams_dattfile.fits
           filtoffsetfile=hxi1_cams_filtoffset.fits
        emapfile=src_exmpmap.fits
        qefile=CALDB
        rmffile=CALDB
        onaxiscfile=CALDB
        onaxisffile=CALDB
           vigfile=CALDB
           outflatfile=hxi1_1_flatfield.fits
        outfile=hxi1_1
        regionfile="@arfgen_region.lis"
        xrtevtfile=src_1_raytrace.fits
           sampling=4.0
```

The example run produces the output files:

```
hxi1_1.arf
hxi1_1.rsp
hxi1_1_flatfield.fits
```

NAME hxisgddtime -- Calculate and correct for deadtime HXI and SGD spectra and lightcurves

USAGE hxisgddtime infile inlcfile inspectfile outlcfile outspecfile gtifile

DESCRIPTION
'hxisgddtime' calculates the deadtime for either HXI or SGD spectra or lightcurves. The task needs the input light curve or a spectral file to be corrected, a file (or a list of files) containing the pseudo events and the GTI file used to produce the input lightcurve and the spectral file. The pseudo-event file (or list of files) is first screened using the science GTI to ensure simultaneity. For the lightcurve correction : the pseudo events are binned with the same bin tine of the science data and for each bin the deadtime is calculated as deadtime= (RATE*TIMEDEL*FRACEXP)/(PSEUDOHZ*TIMEDEL*FRACEXP) where the RATE is count/s in the bin, TIMEDEL is the integration time , FRACEXP is the fractional exposure for each bin and PSUEDOHZ is the frequency with which the PSUESO event are injected in the data stream. The deadtime values are applied to RATE, ERROR and FRACEXP of the science lightcurve and the results stored in the three new columns RATE_CORR, ERROR_CORR, FRAC_CORR.

For the spectra: the deadtime is derived as the total number of events in the pseudo files divided by the PSUEDOHZ*GTI where PSUEDOHZ is the frequency with which the PSUESO event are injected in the data stream and GTI is the sun of the science GTI. The deadtime value is applied to the exposure of the spectrum.

If specified by the parameter merge (default is "yes"), the task then merges the lightcurves and/or spectra.

PARAMETERS

infile [filename]
Input pseudo event file (or @ file list).

inlcfile [filename]
Input lightcurve file (or @ file list).

inspecfile [filename]
Input spectral file (or @ file list).

outlcfile [filename]
Output file root name for lightcurve.

outfile [filename]
Output file root name for spectra and response.

gtifile [filename]
Input science GTI file (or @ file list).

(rspfile = in.rsp) [filename]
Input response file (or @ file list)

(phaseinfo = 400000,1.5,0-1) [string]
Phase filter info in spectra ([epoch,period,phase], NONE)

(merge = yes) [boolean]
Merge output lightcurves or spectra. The default is to merge ([yes]/no).

(expr = none) [string]
Additional expression to select good event or NONE

(mintimedel = 16) [real]
Minimum TIMEDEL for deadtime using pseudo events

(rspweight = 0.5) [string]
Weight factors for addrmf

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
1. Calculates deadtime for HXI and creates a lightcurve and a spectrum
hxisgddtime infile=infile inlcfile=inlcfile inspecfile=inspectfile outlcfile=outlcfile \
outfile=outfile merge=no

NAME hxisgdexpand -- Expand HXI and SGD SFF occurrences to have one signal in each row

USAGE hxisgdexpand infile outfile remapfile badpixfile

DESCRIPTION
'hxisgdexpand' expands the occurrence(s) from an input HXI or SGD SFF such that each output row contains only one signal (for the definition of occurrence and signal, see below). The input file is the output of 'hxisgdpha'.

The output file consists of a copy of the input file with several added columns. For both instruments, the columns called READOUT_ID_INDEX, RECO_STATUS, and PI (energy) are added. In the case of an HXI input file, the task also adds the columns: LAYER, RAWX, RAWY. In the case of an SGD input file, the task only adds one extra column, MATTYPE, containing an integer identifying the originating layer (1 for the Si Layer and 2 for the CdTe layer). The column PI may be populated by either the PHA or the EPI values (see parameter 'expandcol'). If EPI is selected, the value is converted to PI.

A row in the SFF for both HXI and SGD corresponds to an "occurrence" or a possible photon event. Each occurrence contains one or more "signals," each with a given amount of energy deposited in an instrument readout or pixel.

For the task to work, the input file must contain the following columns: TIME, OCCURRENCE_ID, CATEGORY, LIVETIME, STATUS, PROC_STATUS, FLAG_SEU, FLAG_LCHK, FLAG_TRIG, FLAG_TRIGPAT, FLAG_HITPAT, FLAG_FASTBGO, READOUT_ID_RMAP, NUM_ASIC, EPI (or PHA). In addition, for the SGD, the task requires the following additional columns: FLAG_LCHKMIO, FLAG_HITPAR_CC, FLAG_CALMODE, FLAG_CCBUSY

PARAMETERS
infile [filename]
Input filename. This is the output of the tool hxisgdpha.

outfile [filename]
Output filename.

remapfile = CALDB [filename]
FITS file containing the remapping of ASIC and READOUT numbers. If the parameter is set to CALDB, the default, the file is read from the calibration database.

badpixfile = CALDB [filename]
FITS file containing the energy threshold defining a real signal for each readout. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(occurrenceid = -1) [integer]
If greater than zero, the task only expands this single occurrence.

(expandcol = EPI) [string]
Energy column to output in expand mode. If expandcol=PHA, then the signal PHA values are output. If expandcol=EPI, the signal EPI value is converted to PI and output. Any value other than EPI or PHA default to EPI.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Expand all occurrences from the input file to one row per signal and convert EPI to PI
hxisgdexpand infile=HXI_SFF.fits outfile=HXI_output.fits remapfile=CALDB badpixfile=CALDB

2. Expand only occurrence 2 from the input file to one row per signal and write PHA in the PI column.
hxisgdexpand infile=HXI_SFF.fits outfile=HXI_output.fits remapfile=CALDB badpixfile=CALDB occurrenceid=2 expandcol=PHA

NAME hxisgdmerge - Merge slew and pointing raw (unprocessed) event files for the HXI or SGD

USAGE hxisgdpha infile1 infile2 outfile

DESCRIPTION
'hxisgdmerge' clones infile1 to outfile and then appends the rows from infile2 to the end of the new table. This is a very specialized tool that only operates on the raw (SFF) event files from the HXI or SGD, which cannot be merged using 'ftmerge' since they contain data in variable-length array columns. 'hxisgdmerge' does not work on other event files including HXI and SGD event files that have been processed using 'hxievtid' or 'sgdevtid'.

PARAMETERS
infile1 [filename]
First input Second FITS File (SFF) from HXI or SGD to be merged.

infile1 [filename]
Second Input Second FITS File (SFF) from HXI or SGD to be merged.

outfile [filename]
Output file in the Second FITS File (SFF) format.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

1. Merge two files.
hxisgdmerge infile1=sgd_sff_slew.fits infile2=sgd_sff_pnt.fits outfile=sgd_sff_out.fits

---
NAME hxisgdpha - Calibrates the HXI or SGD PHA for each signal in the SFF event file

USAGE hxisgdpha infile outfile gainfile badpixfile

DESCRIPTION
'hxisgdpha' flags events in the STATUS column and computes the energy-proportional pulse-height (EPI) from the PHA column in the SFF event file for HXI and SGD. 'hxisgdpha' checks for each event if they are set bad in the badpixel calibration file using the information stored in the CALDB. For bad events, a flag is set in the STATUS column and the EPI is set to NULL. A flag is set in the STATUS column for SGD events that have the FLAG_LCHK column value set to 1 and the PHA is either 1022 or 1023. For these events the EPI is calculated.
'hxisgdpha' then subtracts the common-mode noise from each PHA (PHA_NSUB hereafter) and if requested outputs the subtracted PHA in the column PHA_NSUB (see parameter 'outnsubco'). If PHA_NSUB is negative a flag is set in the STATUS column, but the event is further processed. The EPI is calculated using gain coefficient stored in a calibration file. If PHA_NSUB is positive but outside of the range of the calibration interval in CALDB, a flag in the STATUS column is set and the EPI is NULL. For the SGD only, there are two gain tables one for trigger in silicon, (normal gain) and the other for the trigger in the CdTe (alternative gain). If an event in the SGD has the FLAG_TRIGPAT set to 1 , 'hxisgdgain' uses the normal gain otherwise uses the alternative gain and a flag is set in the STATUS column.

PARAMETERS
infile [filename]
Input Second FITS File (SFF) from HXI or SGD.

outfile [filename]
Output Second FITS File (SFF). If outfile is the same file as infile, then clobbering must be enabled or the tool exits with an error. Clobbering is enabled by specifying clobber=yes (see below) or by prepending an exclamation point (!) to the output file name.

gainfile = CALDB [filename]
Gain calibration of PHA for the HXI or SGD camera. If the parameter is set to CALDB, the default, the file is read from the calibration database.

badpixfile = CALDB [filename]
Active and bad channels (pixels) file of the HXI or SGD camera. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(outnsubcol = no) [boolean]
Flag to output the PHA_NSUB column. The default is set to no. This column contains the value of the PHA minus the ASIC common mode noise (ASIC_CMN column in the SFF). (yes/[no])

(randomize = no) [boolean]
If set to no (default) does not randomize the fractional part of PHA before calibration. (yes/[no])

(seed = 0) [integer]
Random number generator seed (0 - use system time).

(datamode = NONE) [string]
Substitute DATAMODE in place of event value (or NONE, the default)

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calibrate the PHA, updating the input file in-place:
hxisgdpha infile=sgd_skel_sff.fits outfile=!sgd_skel_sff.fits gainfile=CALDB badpixfile=CALDB

2. Calibrate the PHA, creating a new output file, and adding the PHA_NSUB column:
hxisgdpha infile=sgd_skel_sff.fits outfile=sgd_skel_sff-cal.fits gainfile=CALDB badpixfile=CALDB outnsubcol=yes

---

NAME hxisgdsff - Converts an HXI or SGD First FITS File (FFF) into the Second FITS File (SFF)

USAGE hxisgdsff infile outfile remapfile

DESCRIPTION
'hxisgdsff' unpacks event telemetry data for further processing and reassigns the numbering of the instrument readouts and of the application-specific integrated circuits (ASICs).

Both the FFF or the SFF of an HXI or SGD camera contain a single binary table extension where a row corresponds to an "occurrence" (a possible photon event). An occurrence contains one or more "signals," i.e., deposits of energy in an instrument readout or pixel. The FITS BINTABLE column definition for the FFF and SFF are identical. The empty columns in FFF are filled in the SFF with unpacked telemetry items that are aligned on word or byte boundaries.

In addition to the data format conversion, this tool provides rationalized numbering for the instrument readouts and application-specific integrated circuits (ASICs). The numbering of these components in the telemetry is non-consecutive because it is related to the structure of the instrument electronics. However, consecutive numbers are more convenient for use in subsequent processing steps. The remapping scheme for these numbers is defined by a calibration database file (specified by CALDB or given by the user).

The number of ASICs in an occurrence is stored in the input column named NUM_ASICS. The packed raw telemetry data is stored in the input variable-length vector column called RAW_ASIC_DATA. The output columns containing the unpacked data are called ASIC_ID, ASIC_CHIP, ASIC_TRIG, ASIC_SEU, READOUT_FLAG, NUM_READOUT, ASIC_REF, ASIC_CMN, READOUT_ASIC_ID, READOUT_ID, and PHA. The remapped ASIC and readout numbers are in the output columns ASIC_ID_RMAP and READOUT_ID_RMAP, respectively. All of the output columns are variable-length vector columns.

PARAMETERS
infile [filename]
Input First FITS File (FFF) from HXI or SGD.

outfile [filename]
Output Second FITS File (SFF). May be the same or a different file from infile. If outfile is the same file as infile, then clobbering must be enabled or the tool exits with an error. Clobbering is enabled by specifying clobber=yes (see below) or by prepending an exclamation point (!) to the output file name. Either of these methods cause the file conversion to be done in-place and modify the input file.

(remapfile = CALDB) [filename]
FITS file describing the remapping of ASIC_ID and READOUT_ID to sequential numbers in the columns ASIC_ID_RMAP and READOUT_ID_RMAP. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Convert an FFF to an SFF. The sgd_fff.fits file is copied to sgd_sff.fits, and the output variable-length vector columns in sgd_sff.fits is then filled in with unpacked event data, as described above. The renumbering of instrument components (ASICs and readouts) is obtained from a local file called "ah_sgd_remap.fits".
hxisgdsff infile=sgd_fff.fits outfile=sgd_sff.fits remapfile=ah_sgd_remap.fits

2. Same as above but the renumbering of instrument components (ASICs and readouts) is obtained from the calibration database (CALDB)
hxisgdsff infile=sgd_fff.fits outfile=sgd_sff.fits remapfile=CALDB

NAME hxisgdshield -- Extract lightcurves or spectra from shield data of HXI or SGD

USAGE hxisgdshield infile outfile_root outfile_type datatype columnname hkfile hkext

DESCRIPTION

'hxisgdshield' extracts either light curves or spectra from the shield data of the HXI or SGD.

There are three modes for the output data type: light curve (LC), spectra (SPEC), and 'scan' light curve (SCAN). The mode is selected by the parameter 'outfile_type'. In LC mode ('outfile_type=LC'), the tool generates light curves summed over a given energy range, which is set by the parameter 'energybin'. The parameter 'energybin' may also be set to 'all' to create light curves summed over all energy bins. The total number of energy bins varies according to the input data type. Light curves may be generated from any type of HXI or SGD shield data (see below for a description of the different data types).

In SPEC mode ('outfile_type=SPEC'), spectra are generated by summing the counts within a specified time range for each energy channel. Spectra may only be generated from histogram (HIST) and gamma-ray burst (GRB) data (see below for more information about these data types). The lower and upper bounds of the time range are set by the hidden parameters 'tstart' and 'tstop' respectively. The parameter 'tstart' may be set to 'TSTART', in which case the TSTART keyword of the input extension is used as the lower time bound. Similarly, the parameter 'tstop' may be set to 'TSTOP', in which case the TSTOP keyword of the input extension is used as the upper time bound. Count-type spectra are always created using the same channel definitions as the input file type. In addition, if 'deconvolve=yes' (the default), rate-type deconvolved spectra are also produced with the native instrumental channel definitions (0-4095). In this case, the counts in each channel from the input data are spread uniformly over some number of deconvolved channels, and divided by the exposure to compute the rate.

In SCAN mode ('outfile_type=SCAN'), light curves are generated from the shield data taken during the lower discriminator scan test. This mode may only be used for scalar (SCL) data. The parameters 'tstart' and 'tstop' in SCAN mode have the same meaning and behavior as in SPEC mode. A housekeeping (HK) file is required for SCAN mode. Times from the TIME column in the HK file are used to create a good time interval (GTI) file. This GTI file is used to filter the light curves produced in SCAN mode, and the GTI extension is appended to the final output file. Each GTI in the file is based on a time from the HK file. The GTI START column is determined by subtracting the parameter 'gticut_pre' from each time in the HK file. Similarly, the GTI STOP column is determined by adding the parameter 'gticut_post' to the time in the HK file.

Input Data Types

Both HXI and SGD input shield FITS data come in three types: scalar (SCL) with one value in each input column, histogram (HIST) with 128 channels (numbered 0 to 127), and gamma ray burst (GRB) data with 32 channels (numbered from 0 to 31). The different types of data are stored in different files and/or extensions. This tool analyzes one type of data at a time, thus users must choose the type to analyze using the parameter 'datatype'. Once the user chooses the data type, this tool automatically looks for the input data columns in the appropriate extension(s) of the input data file. FITS input data for all three data types contain multiple columns. This tool creates one output light curve or spectrum FITS file for each input column unless the user specifies particular column(s)for processing using the parameter 'columnname'.

PARAMETERS

infile [filename]

Name of input shield FITS data file.

outfile_root [string]

Initial part of output file name. The name of each column from the input file is appended to this root to create the base name for the output file. The suffix .lc is used for output light curves and .pha for output spectra. For example, if the outfile_root were 'my_data', the output light curve file produced from the input column SH1_FBGO1 would be named my_data_SH1_FBGO1.lc.

outfile_type [string]

Type of output file: LC, SPEC, or SCAN (short for "light curve", "spectrum", or "light curve for scan mode", respectively).

datatype [string]

Data type for which light curve or spectrum is calculated: SCL, HIST, or GRB (short for "scalar", "histogram", or "burst", respectively). SCL is invalid for SPEC mode. Only SCL data is available for SCAN mode.

columnname [string]

Name of column for which light curve or spectrum is calculated. The value may be 'all' or may specify one or more column names, with multiple column names separated by whitespace or commas. If columnname is set to 'all', this tool performs the specified calculations for each input column in the input extension. A separate output FITS file is produced for each input column selected. For example, if columnname were set to 'SH_GRB1' for an HXI gamma ray burst (GRB) light curve, one output file would be produced. If columnname

were 'SH_GRB1, SH_GRB2', two output files would be produced, and if columnname were 'all', six output files would be produced for SH_GRB1 through SH_GRB6.

(energybin) [string]
Input energy bins for HIST or GRB light curves, expressed either as a range of channels, e.g., '3-10' or the word 'all'. This parameter is only used in LC mode. The parameter energybin must be 'all' or 1 for SCL data. HIST and GRB data have 128 channels and 32 channels, respectively. Note that the channel number starts from 0 (zero), so the last channel of HIST data is 127, not 128. To select a single bin, users must still enter a range, e.g., energybin=3-3 to include only bin 3. To include just the first ten channels, users should specify energybin=0-9. When energybin=all, all the channels are used, that is, 0-127 for HIST data and 0-31 for GRB data.

(tstart = TSTART) [string]
Start time in seconds for generating spectra or SCAN light curves. Set to 'TSTART' to use the input extension's TSTART keyword. Used only in SPEC or SCAN mode.

(tstop=TSTOP) [string]
Stop time in seconds for generating spectra or SCAN light curves. Set to 'TSTOP' to use the input extension's TSTOP keyword. Used only in SPEC or SCAN mode.

(deconvolve = yes) [boolean]
If deconvolve=yes, deconvolved spectra with the original 4096 channels (0-4095) are generated in addition to the (convolved) spectra with 128/32 channels from HIST/GRB data. Used only in SPEC mode ([yes]/no).

hkfile [filename]
Input HK file name for GTI search. Required and used only in SCAN mode. In other modes, user may enter any value for this parameter including NONE.

hkext [string]
HK extension name to read. Varies depending on instrument. Required and used only in SCAN mode. In other modes, user may enter any value for this parameter including NONE.

(gticut_pre = 8) [integer]
Duration prior to times in the HK file used to compute the lower limit of each GTI. Times in the HK file indicate the time of the lower discriminator threshold change. Used only in SCAN mode.

(gticut_post = 4) [integer]
Duration after times in the HK file used to compute the lower limit of each GTI. Times in the HK file indicate the time of the lower discriminator threshold change. Used only in SCAN mode.

(cleanup = yes) [boolean]
Indicates whether or not to delete temporary files. Setting to 'no' may help with troubleshooting ([yes]/no).

[ccldhm]

EXAMPLES
1. Create all light curves for HXI1 HIST data with a selection of energy channels.
```
 hxisgdshield.pl infile="HXI1_shield_data.fits" outfile_root="lc" outfile_type="LC" datatype="HIST" columnname=all hkfile=none  \
hkext=none energybin=10-100
```

2. Create a spectrum and a deconvolved spectrum for HXI1 HIST data in the SH_HIST5 column.
```
 hxisgdshield.pl infile="HXI1_shield_data.fits" outfile_root="spec" outfile_type="SPEC" datatype="HIST" \
columnname="SH_HIST5" hkfile=none hkext=none
```

3. Create all scan light curves for SGD1 SCL data.
```
 hxisgdshield.pl infile="SGD1_shield_data.fits" outfile_root="scan" outfile_type="SCAN" datatype="SCL" columnname="all"  \
hkfile="hk.fits" hkext="HK_SGD1_APMU_PRM"
```

NAME mxsgti - Create MXS GTI files

USAGE mxsgti infilehk outfilehk finegti coarsegti timfile delayfile

DESCRIPTION

'mxsgti' creates two GTI files by running 'mxstime' and 'gtiinvert'. First 'mxstime' is run to create the fine and the coarse GTI from the MXS based on parameters from the input SXS HK file. 'gtiinvert' is then run on both the coarse and fine output GTI to create a MXS off GTI for each of the four LED GTIs. Finally, GTI is merged for LED1 & LED3 to create a merged MXS on GTI and merged MXS off GTI. Similarly LED2 & LED4 are merged.

The output fine GTI file has ten extensions:
GTIMXSFNON1
GTIMXSFNON2
GTIMXSFNON3
GTIMXSFNON4
GTIMXSFNOFF1
GTIMXSFNOFF2
GTIMXSFNOFF3
GTIMXSFNOFF4
GTIMXSFNON13
GTIMXSFNOFF24
The output coarse GTI file has ten extensions:
GTIMXSCSON1
GTIMXSCSON2
GTIMXSCSON3
GTIMXSCSON4
GTIMXSCSOFF1
GTIMXSCSOFF2
GTIMXSCSOFF3
GTIMXSCSOFF4
GTIMXSCSON13
GTIMXSCSOFF24

PARAMETERS
infilehk [filename]
Input HK file

outfilehk [filename]
Output HK file

finegti [filename]
Output fine GTI file for MXS

coarsegti [filename]
Output coarse GTI file for MXS

timfile [filename]
Input TIM file

delayfile [filename]
Input instrument delay file (or CALDB)

(leapsecfile = REFDATA) [string]
Input leap second file (or CALDB, [REFDATA])

(stimecol = S_TIME) [string]
Name of S_TIME column

(tioncol = FWE_TI_LED#_ON) [string]
Input TI columns with LED on (#=1-4)

(tioffcol = FWE_TI_LED#_OFF) [string]
Input TI columns with LED off (#=1-4)

(plslencol = FWE_LED#_PLS_LEN) [string]
Input pulse length columns (#=1-4)

(plsspccol = FWE_LED#_PLS_SPC) [string]
Input pulse spacing columns (#=1-4)

(timeoncol = TIME_LED#_ON) [string]
Output LED-on time columns (#=1-4)

(timeoffcol = TIME_LED#_OFF) [string]
Output LED-off time columns (#=1-4)

(calctime = yes) [boolean]
Perform time assignment ([yes]/no)

(calcgti = yes) [boolean]
Produce GTI files ([yes]/no)

(afterglow = no) [boolean]
Add afterglow to fine GTI STOP times (no/[yes])

(dtdecay = CALDB) [string]
Afterglow time [s] (or CALDB)

(interp = twopoint) [string]
Interpolation method (NEAREST, TWOPOINT)

(margingti = yes) [boolean]
Create GTI between TSTART/TSTOP and first/last input GTI

(tstart = DEFAULT) [string]
Value to use for TSTART in seconds (or take from infile)

(tstop = DEFAULT) [string]
Value to use for TSTOP in seconds(or take from infile)

(dt = 0.) [real]
Time separation between input and output GTI (seconds)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
1. The following command calculates GTI for MXS using the default settings for mxsgti, mxstime and gtiinvert:
mxsgti infilehk=sxs_a0.hk1 outfilehk=sxs_a0.hk1.out finegti=fine.gti coarsegti=coarse.gti

NOTES
None, but see help for individual parameters above.

NAME mxstime -- Calculate the times of the MXS start and stop and generates coarse and fine GTI files

USAGE mxstime infile outfile outgti timfile delayfile

DESCRIPTION
The Modulated X-ray Source (MXS) consists of two redundant pairs of LEDs, one direct and one indirect fluorescent source that illuminate, with a specific pulse length and spacing duty, the entire SXS array. The MXS may operate either one or both sources from one pair operating at a time. The telemetry information of the MXS is contained in the extension HK_SXS_FWE of the SXS housekeeping file. 'mxstime' by default always calculates the MXS start and stop times and generates coarse and fine GTI files. The parameters 'calctime' and 'calcgti' may be used to only calculate the start and stop times or only the gti file. 'mxstime' calculates the start and stop times of the four LEDs that are stored in the columns TIME_LED#_ON, TIME_LED#_OFF, where # has values 1 to 4, of the HK_SXS_FWE extension. These times are calculated using: a) the information in the columns TI_LED#_ON, TI_LED#_OFF, b) the header keywords TSTART and TSTOP, c) the column S_TIME, d) the TIM file associated to the observation, e) the caldb files

containing the delay and the leapseconds. Note within a file the TI_LED#_ON/OFF have the same value unless the LED# is turned on or off.

'mxstime' also creates two gti files each containing four extensions, one for each LED. One GTI file records the coarse gtis corresponding to LED start and stop intervals. The coarse GTI output contains the START and STOP columns but also the PLSLEN and PLSSPC columns, recording the pulse length and spacing. Each extension of the coarse GTI also includes two keywords LED#LEN LED#SPC, recording length and spacing. These are set to positive numbers if there is only one pulse length and spacing for all coarse intervals, to negative otherwise and to 0 if that LED has not be turned on. The second GTI file records the fine gti time corresponding to the individual pulses. The fine gtis are calculated using the telemetered information in the columns FWE_LED#_PLS_LEN FWE_LED#_PLS_SPC recording the pulse length and spacing. 'mxstime' by default always add a time to the stop time of each individual pulse (see the parameter 'dtdecay'). By setting the parameter 'afterglow=no' no time is added to the stop time of each pulse. If the 'dtdecay' is large enough to overlap with the next pulse, the gti are merged to avoid creating double exposure. In this case the fine GTI is effectively as the coarse GTI, except for the last stop time that is increased by 'dtdecay' (if the afterglow parameter is set to 'yes'..

For a given LED it is possible that the file only contains TI_LED#_ON or the TI_LED#_OFF value changes, but not both. In this case the GTIs are calculated starting from the TI_LED#_ON, going forward to the end of the file, or from the TI_LED#_OFF, going backward to the start of file, using the length and spacing appropriate for that LED. To all MXS times an offset is added that is specified in the parameter 'mxsoffset'.

PARAMETERS
infile [filename]
Name of the input MXS file. The MXS telemetry information is in the SXS housekeeping file .hk1

outfile [filename]
Name of the output file. This is a copy of the input file where the start and stop time columns for each MXS LED are populated. To overwrite a preexisting file with the same name, prefix the name with an exclamation point '!' (or '\!' on the Unix command line), or else set the 'clobber' parameter = yes.

outgti [filename]
Root of output GTI files. The root name should contain an hash character (#) , ex myfile#, which is replaced by 'cs' and 'fn' words to create two GTI files. Each file has one extension per LED for a total of four extensions.

timfile [filename]
Name of input TIM file relating spacecraft time (TI) to time.

delayfile [string]
Name of input delay file specifying the time delay between the central processor and each MXS. By default this is set to CALDB.

(leapsecfile = REFDATA) [string]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

(stimecol = S_TIME) [string]
Name of S_TIME column.

(tioncol = TI_LED#_ON) [string]
Names of columns in the input file with LED-on TIs (#=1-4).

(tioffcol = TI_LED#_OFF) [string]
Names of columns in the input file with LED-off TIs (#=1-4).

(plslencol = LED#_PLS_LEN) [string]
Names of columns with LED pulse lengths (#=1-4).

(plsspccol = LED#_PLS_SPC) [string]
Names of columns with LED pulse spacings(#=1-4).

(timeoncol = TIME_LED#_ON) [string]
Names of columns to store the output LED-on TIME (#=1-4). Created if not already present.

(timeoffcol = TIME_LED#_OFF) [string]
Names of the columns to store the output LED-off TIME (#=1-4). Created if not already present.

(calctime = yes) [boolean]
If calctime=yes, start and stop times are assigned.

(calcgti = yes) [boolean]
If calcgti=yes, the GTI file is produced.

(afterglow = yes) [boolean]
If afterglow=yes, a time specified in the parameter 'dtdecay' is added to the stop times in the fine gti.

(dtdecay = CALDB) [string]
Time added to the stop time of each pulse (afterglow). By default 'dtdecay' is set to the values stored in CALDB else a time may be specified in seconds.

(mxsoffset = 0.015625) [real]
This time offset is added to all start and stop times.

(interp = twopoint) [string]
Method of interpolation (nearest, twopoint) used when assign time.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Assign time to the input MXS file mxstime mxs_hk_in.fits, writing the results into mxs_hk_out.fits; and, create the coarse (mxs_coarse.gti) and fine (mxs_fine.gti) MXS gti files, using the TIM file timfile.fits and the delay file delay_caldb.fits.
mxstime mxs_hk_in.fits mxs_hk_out.fits mxs_#.gti timfile.fits delay_caldb.fits

---
NAME searchflickpix -- Search for anomalous 'flickering' pixels in event files from CCD-type detectors

USAGE searchflickpix infile outfile cellsize impfac logporb1 logprob2 bthresh

DESCRIPTION
'searchflickpix' is a mission-independent task that reads an input event file to search for flickering pixels, which are pixels that exhibit anomalous pulse height over a short period of time (usually less than an observation) and are caused by radiation damage on a semiconductor detector. The tool outputs a file containing events affected by the detected flickering pixels in the first 'EVENTS' extension. If the parameter 'cleanimg=yes', then the converse is recorded, the clean events not affected by flickering pixels. In the second 'PIXELS' extension, the location of each flickering pixel is recorded, along with the number of counts, duration, and average pulse height of the flickering pixel. This task is based on the ftool 'cleansis', originally developed for the ASCA SIS, but it is significantly altered from the original tool to have more universality and flexibility. It is a mission-independent tool, but is meant to be used for event data from CCD-type detectors, like ASCA SIS and Suzaku XIS.

The main routine for flickering pixel detection is a two-stage process, similar to that in the ASCA task 'cleansis'. This procedure is performed separately for each detector chip as specified by the 'chipcol', 'firstchip', and 'lastchip' parameters. After the event file is read, event counts in each pixel are compared to the mean counts in the full chip, and the pixels where the Poisson probability (calculated using 'impfac') exceeds the threshold given by the 'logprob1' parameter are flagged. The pixel locations for this comparison are defined by the 'xcol' and 'ycol' parameters. Next, the counts in each pixel are compared to the pixels in a surrounding box of size ('cellsize' x 'cellsize'), and anomalous pixels are flagged similarly to the first step but with another Poisson probability parameter, 'logprob2'. If the surrounding pixels yield no counts, a zero background threshold specified by the parameter 'bthresh' is applied instead. This second Poisson search is repeated if 'iterate=yes'. Last, if 'flagedge=yes', pixels along the chip edges are flagged as flickering pixels for the full duration of the observation.

The input events can be filtered several ways before the flickering pixel search is done. By changing 'chanmin' and 'chanmax', a pulse-height filter (<= chanmin && <= chanmax) is applied to the events used for the flickering pixel search. These parameters are ignored if 'chancol=NONE'. Similarly, spatial filters can be applied with 'xmin', 'xmax', 'ymin', and 'ymax'. The default parameter settings uses the TLMIN and TLMAX header keywords corresponding to the appropriate column, effectively applying no filter. A grade filter may be applied using the 'gradecol' and 'grade' parameters. Finally, the 'n_division' parameter may be used to divide the observation into equally-spaced time bins and search each duration separately. This is useful to search for short-duration flickering pixels that might not be flagged in a long observation with high background signal.

The parameter 'cellsize' must be an odd integer or zero, and it is recommended to be about 1/4 of the PSF size, or 7 pixels for SXI. If 'cellsize=0', a simple counts-per-pixel cutoff is applied, so any pixels in the chip image containing more than 'bthresh' counts are flagged as flickering. The recommended value for 'logprob1' is log10(1/chipsize), where 'chipsize' is the total number of pixels in a single detector chip. This is about -5.6 for SXI. For these values, 'bthresh=3' 3 is recommended, although a lower threshold may be set based on the expected background rate.

After the flickering pixels are determined, and as an update from cleansis, this tool calculates and outputs to the 'PIXELS' extension a time duration (or START and STOP) when the flickering events are observed for each detected pixel. This function is disabled when the parameter 'timecol=NONE'. In most cases, START and STOP might be identical to TSTART (observation start time) and TSTOP (observation stop time), respectively. However, given that flickering pixels are not persistent but repeat on timescale ranging from hours to days, the duration can be shorter than the entire observation time, especially for a long-exposure observation. If the parameter 'duration=yes', a detailed duration search is performed for to determine how long each flickering pixel is actually flickering. This information is output as the FRACTION column in the 'PIXEL' extension, and it is the fraction of time the pixel flickers between START and STOP. The time-averaged pulse height in each flickering pixel is also output to the 'PIXELS' extension unless the parameter 'chancol=NONE'.

PARAMETERS
infile [filename]
Input event file. The 'EVENTS' extension must be present.

outfile [filename]
Output event file.

(timecol = TIME) [string]
Time column (or NONE). This column provides the time information used to determine first and last times when flickering events occur in each detected flickering pixel, outputting 'START' and 'STOP' columns in the 'PIXELS' extension. If set to NONE, this time dependence is disabled.

(chipcol = CCD_ID) [string]
Chip column (or NONE). If set to NONE, then all events are searched together for flickering pixels. Otherwise, events are searched separately by 'chipcol' grouping. For an instrument with one physical detector per event list, the parameter should be set to NONE; for instruments with multiple detectors in a single events list, the proper column name should be specified.

(xcol = ACTX) [string]
X coordinate column.

(ycol = ACTY) [string]
Y coordinate column.

(chancol = PI) [string]
Pulse height column (or NONE). The tool calculates the average pulse height values of the flickering events for each detected pixel, and outputs this as the 'chancol' (e.g., 'PI') column in the 'PIXELS' extension. If set to NONE, this feature is disabled and the column is not output.

(gradecol = GRADE) [string]
Event grade column (or NONE). If this name is specified, only events with specific grade(s) given in the 'grade' parameter are used in accumulating the chip image for flickering pixel search. If set to NONE, all grades are used, and the 'grade' parameter is ignored.

(grade = 0) [string]
Event grade(s) for clean (or ALL). This should be a list of grade values to use in accumulating the chip image for the flickering pixel search. Values should be comma-separated, and ranges can be specified by a dash. For example, 'grade=0,2,3,4,6' and 'grade=0,2-4,6' are equivalent. If set to ALL, all grades are used. This is the same result as setting 'gradecol=NONE', and if that is specified, the setting of the grade parameter is ignored.

(n_division = 1) [integer]
Divide total observation time into the given number.

(cleanimg = no) [boolean]
If set to yes, the cleaned (non-flickering) events are output instead of the flickering events (yes/[no]).

cellsize [integer]

Poisson clean cell size (odd integer > 1). This is given in units of pixels, and it must be an odd integer greater than 1 to use that size. If set to 0, this invokes the simple counts-per-pixel cutoff.

impfac [real]
Impact factor for gamma function. This is used for the Poisson search, and in 'cleansis' it has been a fixed value of 320.

logprob1 [real]
The LOG (base 10) of the Poisson probability threshold for the first-step flickering pixel search (using the entire chip). It must be negative. If 'cellsize=0', this parameter is ignored.

logprob2 [real]
The LOG (base 10) of the Poisson probability threshold for the second-step flickering pixel search (comparing to the surrounding pixels). It must be negative. If 'cellsize=0', this parameter is ignored.

(iterate = yes) [boolean]
If set to yes, iterate the second step Poisson clean ([yes]/no).

(flagedge = no) [boolean]
If set to yes, pixels along the chip edge are flagged as flickering for the entire duration of the observation (yes/[no]).

bthresh [integer]
Zero background cutoff threshold, applied when the local mean background level is zero. Check this number against the actual background count rate. If 'cellsize=0', this parameter gives the fixed count cutoff threshold.

(duration = no) [boolean]
If set to yes, a detailed search for flickering duration is performed (yes/[no]).

(sigma = 3.0) [real]
Significance level for flickering duration.

(firstchip = TLMIN) [string]
Minimum value for the 'chipcol' column. If set to TLMIN, the task searches for the header keyword TLMIN of 'chipcol' and uses that value. If set to NONE, this parameter is ignored.

(lastchip = TLMAX) [string]
Maximum value for the 'chipcol' column. If set to TLMAX, the task searches for the header keyword TLMAX of 'chipcol' and uses that value. If set to NONE, this parameter is ignored.

(xmin = TLMIN) [string]
Minimum value for the 'xcol' column. If set to TLMIN, the task searches for the header keyword TLMIN of the related column, and uses that value.

(xmax = TLMAX) [string]
Maximum value for the 'xcol' column. If set to TLMAX, the task searches for the header keyword TLMAX of the related column, and uses that value.

(ymin = TLMIN) [string]
Minimum value for the 'ycol' column. If set to TLMIN, the task searches for the header keyword TLMIN of the related column, and uses that value.

(ymax = TLMAX) [string]
Maximum value for the 'ycol' column. If set to TLMAX, the task searches for the header keyword TLMAX of the related column, and uses that value.

(chanmin = TLMIN) [string]
Minimum value for 'chancol' to use in accumulating the chip image for flickering pixel search. If set to TLMIN, the task searches for the header keyword TLMIN of the related column, and uses that value. If 'chancol=NONE', this parameter is ignored.

(chanmax = TLMAX) [string]
Maximum value for 'chancol' to use in accumulating the chip image for flickering pixel search. If set to TLMAX, the task searches for the header keyword TLMAX of the related column, and uses that value. If 'chancol=NONE', this parameter is ignored.

[ccldhm]

EXAMPLES
1. Detect flickering pixels from an SXI event file using a 5x5 background cell and a 3 count-zero background threshold. The output file contains the "flickering pixels" (not cleaned events, as was output by cleansis) in the 'EVENTS' extension.
searchflickpix infile="input.evt" outfile="output_flickering.evt" cellsize=7 impfrac=320 logprob1=-5.6 logprob2=-5.6 bthresh=3

2. Same as above, but write out the cleaned (non-flickering) event list in the output file.
searchflickpix infile="input.evt" outfile="output_clean.evt" cellsize=7 impfrac=320 logprob1=-5.6 logprob2=-5.6 bthresh=3 \
cleanimg=yes

3. Same as example 1, but a Suzaku XIS event file is used as input. Each XIS event file contains data from one CCD, but with a larger chip size (1024x1024).
searchflickpix infile="input_xis.evt" outfile="output_flickering.evt" cellsize=5 impfrac=320 logprob1=-6.0 logprob2=-6.0 bthresh=3 \
chipcol=NONE xcol=DETX ycol=DETY

4. Clean an event file to global fixed threshold of 20 counts per pixel. Exclude PHA values greater than 1200.
searchflickpix infile="input.evt" outfile="output_clean.evt" cellsize=0 impfrac=320 logprob1=-6.0 logprob2=-6.0 bthresh=20 \
chancol=PHA chanmin=0 chanmax=1200 cleanimg=yes

NAME sgdarfgen -- Calculate SGD transmission ratio for a given source and correct the rsp for transmission

USAGE sgdarfgen infile rspfile outfile ra dec sgdid ccid

DESCRIPTION
'sgdarfgen' computes the SGD transmission ratio for a source on the sky, given a satellite pointing, and corrects the response file for transmission. The transmission ratio depends on source position because the SGD collimator partially blocks light from off-axis sources. 'sgdarfgen' requires the source position, specified in the parameters 'ra' and 'dec', the pointing read from the keywords header RA-NOM/DEC_NOM/PA_NOM of the spectral file (see parameter 'infile') and the on-axis response from CALDB (see parameter 'rspfile'). 'sgdarfgen' calculates transmission ratio for each Compton Camera(CC) or for the combined CC in one SGD. If the parameter 'ccid' is set to either 1 or 2 or 3, the transmission is calculated for a single CC. If the parameter 'ccid' is set to either 1,2 or 2,3 or 1,3 the transmission is calculated for the two CC specified. If the 'ccid' is set to 0, the transmission is calculated for all three CC in an SGD. If 'cccid' is set to 1 or 2 or 3 only, one RSP is required as input to match the SGD CC combination. If 'ccid' is set to 3, three input RSP files are required one for each of the CC for the defined SGD. The transmission computation is done using the CALDB transmission file ('tranratfile') by interpolating between points in FOC coordinates for any energy in the RSP energy grid. The outputs are the following : one or three transmission files and the corresponding RSP depending on the 'ccid'. The root name for the transmission and RSP are specified in the parameter 'outfile'.
This task should be used for sources off-axis. The RSP in CALDB is for on-axis sources and therefore there is not need to use this task to make a correction.

PARAMETERS
infile [filename]
Name of input spectral file from which nominal values of RA, Dec, and roll angle are read.

rspfile [filename]
Name of input RSP file(s) or file list. If the parameter 'ccid' is not 0 (i.e., either 1, 2, or 3,) then a single file name should be specified. If the parameter 'ccid' is 0, then three file names separated by commas or a file list including three file names should be given. The DETNAM keyword in the given file(s) must be consistent with the 'ccid'.

outfile [filename]
Name root of the transmission and RSP files. The output filename is constructed using the root and appending the SGD number from 'sgdid' and the CC from 'ccid' parameters. For example for SGD1 CC1 is root_sgd1_cc1.rsp and root_sgd1_cc1.arf, if sgdid=1 and ccid= 0 six files are created as root_sgd1_cc1.rsp and root_sgd1_cc1.arf root_sgd1_cc2.rsp and root_sgd1_cc2.arf root_sgd1_cc3.rsp and root_sgd1_cc3.arf

ra [real]
Right Ascension of the source. [deg]

dec [real]
Declination of the source. [deg]

sgdid [integer]
SGD unit ID (1 or 2).

ccid [integer]
Compton camera ID (0, 1, 2, or 3: 0 for all).

(tranratfile = CALDB) [filename]
Input transmission ratio file. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Create an ARF file of SGD1/CC1 for a source at RA = 214.0 deg and Dec = 36.0 deg.
sgdarfgen infile=input.pi rspfile=sgd1_cc1.rsp outfile=out_sgd1_cc1 ra=214.0 dec=36.0 \
sgdid=1 ccid=1

2. Create ARF files of all CCs of SGD2 for a source at RA = 214.0 deg and Dec = 36.0 deg.
sgdarfgen infile=input.pi rspfile=sgd2_cc1.rsp,sgd2_cc2.rsp,sgd2_cc3.rsp \
outfile=out_sgd1 ra=214.0 dec=36.0 sgdid=2 ccid=0

NAME sgdevtid - Reconstruct SGD events

USAGE sgdevtid infile outfile remapfile fluorefile badpixfile probseqfile probfovfile

DESCRIPTION
'sgdevtid' determines whether the signals in an occurrence constitute a valid event and computes the event energy and the 3-dimensional coordinates of its first interaction.

The input file to 'sgdevtid' is an SFF file with variable length array columns where each row contains an occurrence with multiple signals recorded at a given time. The output event file, specified by the parameter 'outfile', has a different structure and does not contain columns with variable length arrays. Each row in the output file corresponds to a reconstructed event and the columns 'CAMERAX', 'CAMERAY', 'CAMERAZ', and 'PI' contain the reconstructed position and energy of that event along with additional columns containing information related to the number and the type of interactions (see below).

'sgdevtid' has many parameters that are used in the event reconstruction. These have been optimized and it is advised to use their default settings.
The algorithm uses the different signals in each occurrence to reconstruct an event in a two-step process. The first step is to merge the signals according to their location and whether or not are consistent with fluorescence X-ray. The second step is to analyze the remaining signals and determine whether the sequence is consistent with an event. If the algorithm fails, the value of 'PI' is set to 'NULL ‘and writes diagnostic columns the reason to why the occurrence could not be reconstructed. Below is a more detail description of the algorithm.

The merging of the different signals depends on what is considered the maximum distance for these signals to be physically related. This critical distance in turn depends on the event geometry. The task first merges the multiple adjacent signals (charge sharing) present in a single detector layer. The parameter 'd10' defines the meaning of "adjacent" and specifies the critical distance for the merging to occur. The next step of the task consists in merging the fluorescence signals into the originating signal. This step depends critically of the event configuration. If both the originating and fluorescence signals are in the same CdTe layer, the critical distance is specified by the parameter 'd1a1a', if not, the relevant parameter is called 'd1a1b'. If the originating signal is in CdTe but the fluorescence signals are detected in a Si layer, the critical distance is now specified by the parameter 'd1a2'. If both are in the Si layers, the relevant parameter for the merging becomes 'd1a3'. These parameters are hidden in the task and a value recommended by the instrument team is assigned by default.

At this point (one step/merger), the task reconstruction has reached the level of "hit". If there is only one hit, the process is done. Otherwise, the task now reconstructs the different "hits" until the process converges on a single most-likely sequence of hits.

The following step depends on the number of reconstructed hits. If there are more than 4 hits, the entire occurrence, considered consistent with a cosmic ray, is rejected. If there are between 2 and 4 hits, all the possible permutations for the sequence of hits are tried and all sequences with non-physical Compton scattering (F-test) are rejected. If only one sequence remains, it is deemed to be the

actual reconstructed event. If there are more than one sequence at the end of the first test, subsequent tests are performed. If the sequences have inconsistent Compton scattering angle and geometric scattering angle (for 3 or 4 hits; G test) are rejected. Uncertainties in geometric scattering angle are computed using the method specified by the parameter 'delgmethod' (either set to 'analytic' or 'corner'). The maximum acceptable discrepancy between geometric and Compton scattering angle is given by the uncertainty in scattering angle (computed with the method specified above) multiplied by the value of the parameter 'b' (set to '1.0' by default). Again, if only one sequence remains, the process is done and the sequence is declared the reconstructed event. If all sequences are rejected, the task calculates the escape energy, the unabsorbed part of the energy of a photon that is able to exit the camera after detection, and executes the previous F and G tests (for 3 or 4 hits) again, correcting for that effect. A calibration file specified by the 'probseqfile' parameter allows to abandon the low-probability sequences. The probability thresholds for 2, 3, and 4 hits are given by parameters probaccept2, probaccept3, and probaccept4, respectively.

The last step is to rank the remaining sequences by a figure of merit (FOM). The figure-of-merit is an angular resolution measure comparing the calculated kinematic scattering angle to the geometrical scattering angle, which is calculated by assuming that the incident direction is the line of sight.
The FOM can be tuned using the offset parameters paraoffset0, paraoffset1, paraoffset2, and the weighting parameters weight0, weight1, weight2, and weight3. The highest-ranking sequence is then declared the reconstructed event.
The FOM formulae are given below.
If M=2, FOM[k] = (paraoffset0 - G[k,0]) * weight0 + Prob[k] * weight3
If M=3, FOM[k] = (paraoffset0 - G[k,0]) * weight0 + (paraoffset1 - G[k,1]) * weight1 + Prob[k] * weight3
If M=4, FOM[k] = (paraoffset0 - G[k,0]) * weight0 + (paraoffset1 - G[k,1]) * weight1 + (paraoffset2 - G[k,2]) * weight2 + Prob[k] * weight3
M is the number of hits, G[k,n] is the difference between the kinematic angle and the geometric angle for hit n, and Prob[k] is read from the calibration file specified by the 'probseqfile' parameter.

Reconstruction results for each occurrence are in the output file specified by the 'outfile' parameter.
Columns copied from the input SFF are TIME, OCCURRENCE_ID, CATEGORY, FLAG_LCHKMIO, FLAG_CCBUSY, FLAG_HITPAT_CC, FLAG_HITPAT, FLAG_FASTBGO,
FLAG_SEU, FLAG_LCHK, FLAG_CALMODE, FLAG_TRIGPAT, FLAG_TRIG,
LIVETIME, PROC_STATUS, and STATUS.
Output file columns populated by the reconstruction are as follows:
PI: pulse invariant
ENE_TOTAL: sum of EPI per occurrence
NUMSIGNAL: number of signals in occurrence
NUMHITS: 5-element array giving hit distribution; default value of each element is zero; element N (N=1,2,3, or 4) is populated with 1 if there are N or more hits; element 5 is populated with 1 if
escape energy is calculated
SEQ_HITS: descriptor for hit mechanism, from CALDB file given by 'probseqfile' parameter
DELCOMPTON: value of Delta G (M>2) for first two hit pairs = difference between cosines of Compton and kinematic angles
COMPTON_TH: [deg] Theta_K(0) = Compton scattering angle
COMPTON_PH: [deg] azimuthal angle of second hit with respect to first hit
DISTANCE0: [mm] physical distance between first two hits
OFFAXIS: [deg] difference between Compton and kinematic angles
CAMERAX, CAMERAY, CAMERAZ: [mm] 3-dimensional coordinates of first hit within the Compton camera
LIKELIHOOD: likelihood of the event
RECO_STATUS: bit flags describing the reconstruction of each occurrence (see below)
MATTYPE: code for the material where the incident photon is detected = 1 for Si, 2 for CdT2, or 3 for multiple layers

The bit flags in the RECO_STATUS (reconstruction status) column are as follows:

Bit	Description
	------(Occurrence mode identification)----------
0	Mode is Am 241
1	Mode is PSEUDO
2	Mode is CALMODE
3	Mode is READALL
4	Mode is NOTSURE
	------(Event reconstruction skipped for this occurrence)----------
5	Bad PROC_STATUS
6	Mode is READALL or CALMODE and SKIPRECO option is YES
7	Occurrence has no signals
8	FASTBGO or HITPAT flag is set and REJECTBGO option is YES
	------(Event reconstruction outcome)----------
9	Event could not be reconstructed

Bit Description
------(Occurrence mode identification)------------------------------
0 not used
1 Mode is PSEUDO
2 Mode is CALMODE
3 Mode is READALL
4 Mode is NOTSURE
------(Event reconstruction skipped for this occurrence)-------------
5 Bad PROC_STATUS
6 Mode is READALL or CALMODE and SKIPRECO option is YES
7 Occurrence has no signals
8 FASTBGO or HITPAT flag is set and REJECTBGO option is YES
9 Occurrence has too many signals to reconstruct
10 All signals in the occurrence are below threshold
------(Event could not be reconstructed)----------------------------
11 Signal cluster has too many signals
12 Signal cluster has invalid shape
13 Too many signals in a group within Si layers
14 Occurrence has too many hits to reconstruct
15 All hits are bad by F test for Compton scattering (2 or fewer hits)
16 All hits are bad by F test for Compton scattering (3 or more hits)
17 All hits are bad by G test for Compton scattering
18 All hits are bad because they have low probability (2 or fewer hits)
19 All hits are bad because they have low probability (3 or more hits)
20 Singularity in escape energy computation
------(Trivial reconstruction)---------------------------------------
21 One signal above threshold, and one signal in the occurrence
22 One signal above threshold, and multiple signals in the occurrence
------(Complex reconstruction succeeded)----------------------------
------(reconstruction by merging signals into one hit)-------------
23 One hit is left after merging signal clusters
24 One hit is left after merging CdTe fluorescence within a CdTe layer
25 One hit is left after merging CdTe fluorescence from a different CdTe layer
26 One hit is left after merging CdTe fluorescence from Si layer
27 One hit is left after merging scattered electron energy from Si layer to Si layer
------(reconstruction by evaluating possible hit sequences)--------
28 One hit sequence is permitted by F test (cosine of scattering angle valid)
29 One hit sequence is permitted by G test (scattering angle consistent with energies)
30 One hit sequence is permitted by probability threshold
31 One hit sequence is left after tie-breaking based on figure of merit (FOM)
32 not used
------(Reason for escape energy calculation, if one was done)--------
33 Because all sequences initially ruled out by F test
34 Because all sequences initially ruled out by G test
35 Because all sequences initially ruled out by probability test
------(Whether escape energy was calculated)-------------------------
36 Escape energy calculation was done
37 not used
38 not used
39 not used

PARAMETERS
infile [filename]
Name of the SGD input file. This should be the SFF, output of the 'hxisgdpha' task.

outfile [filename]
Name of the output event file.

remapfile = CALDB [filename]
Name of the file containing the remapping of ASIC and READOUT numbers. If the parameter is set to CALDB, the default, the file is read from the calibration database.

fluorefile = CALDB [filename]
Name of the file containing the energy ranges of fluorescence electrons in CdTe and Si. If the parameter is set to CALDB, the default, the file is read from the calibration database.

badpixfile = CALDB [filename]
Name of the file containing the energy threshold defining a real signal for each readout. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(probseqfile = CALDB) [filename]
Name of the file assigning the probability of each possible sequence of interactions. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(probfovfile = CALDB) [filename]
Name of the file containing the probability for photons to be in the SGD camera field of view, based on incident angle. If the parameter is set to CALDB, the default, the file is read from the calibration database.

(occurrenceid = -1) [integer]
If greater than zero, the task only expands this single occurrence.

(rejectbgo = NO) [boolean]
Reject events (the default) for which BGO trigger occurred. (yes/[no])

(skipreco = NO) [boolean]
By default skip the reconstruction of READALL and CALMODE events. (yes/[no])

(strangepix = 0) [integer]
Treatment of strange pixel ([0]=bad, 1=normal, 2=strange). Option 2 is provisional and not implemented.

(outtracefile = NONE) [filename]
Output a file containing the reconstruction information for each occurrence by detector side.

(numsignal = 48) [integer]
Maximum number of signals to analyze. Reconstruction is not attempted on occurrences with more than this number of signals.

(d10 = 3.2) [real]
Orthogonal distance (in mm) between two adjacent pixels in the same detector layer; used in identifying charge sharing events.

(d1a1a = 9.0) [real]
Diagonal distance (in mm) between two adjacent pixels in same detector layer; used in identifying possible CdTe-CdTe fluorescence signals in a single layer.

(d1a1b = 5.0) [real]
Distance (in mm) between two CdTe layers; used in identifying possible CdTe-CdTe fluorescence signals in different layers.

(d1a2 = 14.0) [real]
Distance (in mm) between CdTe and Si layers; used in identifying possible Si-CdTe fluorescence signals.

(d1a3 = 5.0) [real]
Distance (in mm) between Si and Si layers; used in identifying possible electron scattering signals.

(b = 1.0) [real]
Acceptance (in mm) tolerance for G test as described above.

(probaccept2 = 0.5) [real]
Probability acceptance threshold for 2 hits as described above.

(probaccept3 = 0.5) [real]
Probability acceptance threshold for 3 hits as described above.

(probaccept4 = 0.5) [real]
Probability acceptance threshold for 4 hits as described above.

(distz = 1000000.0) [real]
[mm] Very large distance for target object as described above. The Z-direction is toward the sky, so this distance represents the distance to an object far away from the detector

(paraoffset0 = 1.6) [real]
Offset parameter combined with G[k,0] for the figure-of-merit (FOM) discrimination as described above.

(paraoffset1 = 1.0) [real]
Offset parameter combined with G[k,1] for the figure-of-merit (FOM) discrimination as described above.

(paraoffset2 = 1.0) [real]
Offset parameter combined with G[k,2] for the figure-of-merit (FOM) discrimination as described above.

(weight0 = 1.0) [real]
Parameter used in weighting G[k,0] for the figure-of-merit (FOM) discrimination as described above.

(weight1 = 1.0) [real]
Parameter used in weighting G[k,1] for the figure-of-merit (FOM) discrimination as described above.

(weight2 = 1.0) [real]
Parameter used in weighting G[k,2] for the figure-of-merit (FOM) discrimination as described above.

(weight3 = 1.0) [real]
Parameter used in weighting Prob[k] for the figure-of-merit (FOM) discrimination as described above.

(delgmethod = ANALYTIC) [string]
Method to be used to calculate the error in the test of consistency between geometric and kinematic scattering angles. ([ANALYTIC]|CORNER)

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Reconstruct events using the default parameters; Do not output any additional information.
sgdevtid infile=SGD_FFF.fits outfile=SGD_output.fits remapfile=CALDB fluorefile=CALDB badpixfile=CALDB \
probseqfile=CALDB probfovfile=CALDB

2. Reconstruct events using some user-defined parameters to override the defaults, output any additional information, and expand output to include one row per signal.
sgdevtid infile=SGD_FFF.fits outfile=SGD_output.fits remapfile=CALDB fluorefile=CALDB badpixfile=CALDB \
probseqfile=CALDB probfovfile=CALDB d1a_1a=9.5 prob_accept2=0.6 paraoffset0=2.6 weight2=0.9\
outtracefile=SGD_extra_output.fits clobber=yes

NAME sgdgainfit -- Calculate the SGD time-dependent energy gain corrections for events from comparison with known calibration lines

USAGE sgdgainfit infile outevtsuffix outfile

DESCRIPTION
'sgdgainfit' is a script that runs 'ahgainfit' on SGD event files and allows the user to select the events input into 'ahgainft'. The description of 'ahgainfit' and its parameters also used by 'sgdgainfit' are described below.

(a) If 'rungtitgen=yes', 'sgdgainfit' creates a GTI file. The GTI are created using the expression either specified in the parameter 'gtiexpr' and/or using the default expression label ('gtigenlabel' parameter) stored in a CALDB file ('selectfile' parameter). The expression label refers to entries listed in the input mkf file ('gtigen_infile' parameter). If the 'gtifile' parameter is set, the single GTI file or a list of GTI files specified is also used.

(b) If 'runscreen=yes', 'sgdgainfit' screens the input event data with the expression specified in the parameter 'expr' and/or using the default expression label ('screenlabel' parameter) stored in the CALDB file specified by the 'selectfile' parameter.

'sgdgainfit' creates two outputs. The first is a gain correction file identical in structure to that produced by 'ahgainfit'. The second is the screened event file created within 'sgdgainfit'. The filename of the screened event file is the same as the input file with a suffix (see 'outevtsuffix' parameter) appended.

ahgainfit:
'ahgainfit' calculates time-dependent energy gain corrections by comparing the theoretical and observed energies of a calibration line or line complex. For each run of the task, only one line may be specified to calculate the gain correction (see parameter 'linetocorrect'). 'ahgainfit' is a general task that is used directly in the scripts 'sxigainfit', 'hxigainfit', and 'sgdgainfit' for the SXI, HXI, and SGD respectively. For the SXS, the 'sxsgain' tasks uses the same core fitting function of 'ahgainfit'.

'ahgainfit' takes as input an event file with time and energy columns, and requires that the events are time ordered. The task accumulates spectra from events that are consecutive in time, with energy centered on the calibration feature and compares each spectrum from the events with a theoretical model of the calibration feature profile. The calibration feature used by 'ahgainfit' is specified by the parameter 'linetocorrect' as a string, and the names and energies of the features are specified in a calibration file (parameter 'linefitfile'). The calibration feature may be composed of many atomic or nuclear line components that are listed in the calibration file.

The energy range for the spectra constructed from the event file as well as from the theoretical profile, may be specified in two different ways. (1) The default energy range for the spectra is the smallest and largest energies of the line components, and expanded with the 'extraspread' parameter, i.e. [E_min - extraspread : E_max + extraspread]. It is recommended to set 'extraspread' larger than the sum of the natural width of the calibration feature, the value of the 'broadening' parameter, and the magnitude of the expected energy shift. (2) Alternatively, the energy range may be specified by setting the 'startenergy' and 'stopenergy' parameters. If these parameters are non-negative 'ahgainfit' uses their values to accumulate the spectra, instead of the range derived using the 'extraspread' setting. The energy column used to accumulate the spectra is specified by the 'energycol' parameter. The energy column is expected to be in units of channel, where the eV per channel is set by the 'evchannel' parameter. The spectra are binned according to the 'binwidth' parameter, where the 'binwidth' is in units of 'energycol' channels. The theoretical profile is constructed on a mesh defined by this energy range and 'binwidth', where each calibration line is assumed to be Lorentzian. The profile may be convolved with a Gaussian having the FWHM given by the 'broadening' parameter.

The number of events in each spectrum is defined by the 'numevent' and 'minevent' parameters. The task accumulates spectra with a number of events between 'minevent' and 'numevent'. However, if a spectrum has fewer than 'minevent' points, then it is combined with the previous spectrum if possible. Therefore all spectra have a size between 'minevent' and ('numevent+minevent-1'). To avoid having spectra accumulated over large gaps in time, the group of points in the spectrum is truncated when the time interval between consecutive events is greater than the 'gapdt' parameter. Adjacent spectra in time may share a percentage of their points based on the 'grpoverlap' parameter that may vary between 0 and 100. If 'grpoverlap' is set to 0, the consecutive spectra share no points in common; if set to 100 they share all points in common but one.

The spectra events may be simultaneously collected based on the value of a column present in the event file, given by the 'splitcol' parameter. This option may be used, for example, to find the gain correction for each layer of the SGD detector ('splitcol=LAYER'). If a GTI file is specified by the 'gtifile' parameter, events outside of these GTI intervals are excluded. Spectra are not accumulated across GTI intervals unless the 'spangti' parameter is set to 'yes'.

For each accumulated spectrum, 'ahgainfit' fits the theoretical profile to the data and also derives binned and unbinned averages. A least-squares method is used in the fitting. The fitted parameters are energy shift, scaling factor, background (unless the 'background' parameter is set to NONE), and, optionally, convolution width if 'fitwidth=yes'. The background is fit with a constant value if 'background' is set to CONSTANT, and a power-law if set to SLOPE. The unbinned average energy (as specified by 'energycol') is the sum of the energies in the spectrum divided by the number of events in the spectrum. The binned average energy is the weighted average derived by summing, over bins in the spectrum, the product of the energy and number of events per bin, and then dividing by the total number of events in the spectrum. The fitted gain corrections is computed from the fitted shift with respect to the theoretical line profile. The binned average gain correction is computed from the difference between the profile and spectrum averages.

The default values for the parameters used in the fitting method ('minwidth0', 'maxitcycle', 'r2tol', 'searchstepshift', 'maxdshift', 'bisectolshift', 'searchstepwidth', 'maxdwidth', 'bisectolwidth', and 'minwidth') should not need to change since already optimized.

The output file has two extensions. One extension, GRID_PROFILE, contains the energies and amplitudes of the theoretical profile used in the fitting procedure, including any convolution from the 'broadening' parameter. The other extension, DRIFT_ENERGY, reports the fitting results for each spectra in the following columns: TIME (midpoint of the time interval over which the spectrum is collected), splitcol (value of splitcol for spectrum as given by the 'splitcol' parameter; this column is absent if 'splitcol=NONE'), COR_FIT (energy correction factor from spectrum fit), COR_AVE (energy correction factor from spectrum average), CHISQ (reduced chi-squared of the

fit), AVGUNBIN (average energy of events in spectrum prior to binning), AVGBIN (weighted spectrum average energy), AVGFIT (average energy from fit), SHIFT (fitted energy shift), SCALE (fitted vertical scaling factor), BGRND (fitted background), WIDTH (if 'fitwidth=no', same as broadening parameter; if 'fitwidth=yes', fitted width), TELAPSE (difference between times of first and last event in spectrum), EXPOSURE (calculated using the GTI), NEVENT (total number of events collected for this spectrum), BINMESH (array containing the count spectrum energy bins), SPECTRUM (array containing the observed binned count spectrum), FITPROF (array containing theoretical profile with fitted parameters applied). If the 'calcerr' parameter is set to 'yes', one-sigma errors for the SHIFT and WIDTH are calculated. The errors are calculated with chi-squared and maximum-likelihood methods and output in the columns SIGSHCHI2, SIGWDCHI2, SIGWDLIKE and SIGWDLIKE respectively. If 'writeerrfunc' parameter is set, the chi-squared and likelihood calculated valued are output in the arrays SHCHI2, SHLIKE, WDCHI2 and WDLIKE. The numbers of values output in these arrays are specified in the 'nerrshift' and 'nerrwidth' parameters, respectively.

PARAMETERS
infile [filename]
Input event file name. Input file may be a single file or a list of files (the name of the text file with the list preceded by the "@" character).

outevtsuffix [string]
Screened output file name suffix. This string is appended to the input file name (or name of first input file if a list).

outfile [filename]
Output gain correction file name.

(rungtigen = no) [boolean]
If set to yes, run ahgtigen to create GTI (yes/[no]).

(runscreen = no) [boolean]
If set to yes, run ahscreen to filter input file (yes/[no]).

(gtigen_infile = mkf.fits) [filename]
Input mkf file used to create a GTI used in screening. This parameter is ignored if rungtigen is set to no.

(gtifile = NONE) [string]
Input gti file, or name of text file with list of files, preceded by "@" if including multiple input gti files to be merged. If the parameter is set to NONE no input GTI file is used. This parameter is ignored if rungtigen is set to no.

(selectfile = NONE) [string]
CALDB or user input label file with labels and expressions. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used.

(gtigenlabel = NONE) [string]
Labels to read from label file specified by selectfile and applied to gtigen_infile to create GTI, or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if rungtigen is set to no.

(screenlabel = NONE) [string]
Labels to read from label file specified by selectfile and used to screen input event file(s), or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if runscreen is set to no.

(gtiexpr = NONE) [string]
Expression, or label to read from label file specified by selectfile, used to create GTI. If the parameter is set to NONE no expression is used. This parameter is ignored if rungtigen is set to no.

(expr = NONE) [string]
Additional event screening expression used to screen input event file(s). If the parameter is set to NONE no expression is used. This parameter is ignored if runscreen is set to no.

(linefitfile = line.in) [filename]
Input CALDB file containing parameters of the Lorentzian components used to construct the theoretical line profile. If the parameter is set to CALDB, the file is read from the calibration database.

(linetocorrect = MnK) [string]
Calibration line to use for the gain correction. The value must match an extension in linefitfile. For the SGD the relevant lines are 103mRh, 127mTe, 129mTe, 125mTe, 123Te, 115mIn, and the 511 keV annihilation feature.

(energycol = PI) [string]
Name of energy column to use in gain fitting.

(splitcol = SIDE) [string]
Column name used to separate the data.

(numevent = 1000) [integer]
Number of events collected for each spectrum used to calculate a single gain correction.

(minevent = 750) [integer]
Minimum number of events required for a spectrum. If the length of a group is less than minevent, those points are included with the previous group for processing, if possible.

(gapdt = -1.) [real]
The upper limit to the time interval between two consecutive events in the same spectrum used in fitting. Two consecutive events separated in time by more than this amount are assigned to different groups. If gapdt=-1, no limit is imposed.

(grpoverlap = 0.) [real]
The percentage overlap between adjacent groups. For grpoverlap=100 adjacent groups are shifted by one event, for grpoverlap=0 adjacent groups are independent and share no events.

(startenergy = -1) [real]
Beginning of energy range in eV over which the spectra are collected. If startenergy is negative, the first energy is automatically determined by the smallest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(stopenergy = -1) [real]
End of energy range in eV over which the spectra are collected. If stopenergy is negative, the final energy is automatically determined by the largest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(extraspread = 500) [real]
Energy in eV by which the energy range is extended on either side beyond the smallest and largest energy in the linefitfile for the selected calibration feature. This parameter may be overridden by the startenergy and stopenergy parameters.

(evchannel = 100) [real]
Conversion factor from channel number (for energycol) to energy [eV/chan].

(binwidth = 1.0) [integer]
Energy bin width, in units of channels, to use when collecting spectra.

(broadening = 200) [real]
FWHM of the Gaussian in eV used to initially broaden the theoretical line profile. If fitwidth is set to no, the profile width is fixed at this value.

(gridprofile = no) [boolean]
If gridprofile is set to yes, only output the theoretical profile including any convolution due to the broadening parameter; no fitting is conducted (yes/[no]).

(fitwidth = yes) [boolean]
If fitwidth is set to yes, then fit the width of each spectra in addition to the energy shift (yes/[no]).

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE).

(spangti = yes) [boolean]
If spangti is set to yes, events in different intervals in gtifile may be collected in the same spectrum to be fit. If spangti is set to no, then groups of events used to construct the spectra must be from the same GTI. This parameter is ignored if gtifile is set to NONE (yes/[no]).

(avgwinrad = -1.) [real]
Radius of interval (in units of binwidth) used only to update the initial shift estimate prior to fitting. If avgwinrad is set to -1, this radius is automatically calculated based on the theoretical line profile and the broadening parameter. This is not used in calculating the average results.

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no]).

(writeerrfunc = no) [boolean]
Output the array of chi-squared and likelihood calculated for the SHIFT and WIDTH (yes/[no]).

(minwidth0 = 1.0) [real]
Smallest width, in units of binwidth, allowed as the initial value in width fitting. This parameter provides a lower limit to the initial estimate of the width as computed by the fitting algorithm. The value must be greater than zero.

(maxitcylce = 5) [integer]
Maximum number of fitting iterations.

(r2tol = .01) [real]
Convergence criterion on R-squared for least-squared fitting. Once R-squared changes by less than this amount between fitting iterations, the procedure is finished. This parameter should not normally need to be changed from the default value.

(searchstepshift = 2.) [real]
Step size, in units of binwidth, used when searching for best-fit energy shift in either direction from the initial shift estimate based on the spectrum average. The final shift is obtained using the bisection method (see bisectolshift).

(maxdshift = 5.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of energy shift. If no solutions are found within this deviation at smaller or larger shifts from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolshift = .1) [real]
When the bisection method determines the energy shift to within this amount in units of binwidth, the fitting procedure is completed.

(searchstepwidth = 5.) [real]
Step size, in units of binwidth, used when searching for best-fit convolution width in either direction from the initial width estimate based on the difference between the profile and spectrum statistical variances. The final width is obtained using the bisection method (see bisectolwidth).

(maxdwidth = 10.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of convolution width. If no solutions are found within this deviation at smaller or larger widths from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolwidth = .2) [real]
When the bisection method determines the convolution width to within this amount in units of binwidth, the fitting is procedure is completed.

(minwidth = .5) [real]
Since the least-squares fitting functional is undefined when the width is zero, one must define a minimum allowed fitted width. If the fitting routine attempts to fit a width smaller than this value (in units of binwidth), the fitting procedure fails for the spectrum.

(nerrshift = 100) [integer]
Number of shift values in uncertainty calculations.

(nerrwidth = 100) [integer]
Number of width values in uncertainty calculations.

(shifterrfac = 3.0) [real]
Factor for determining domain of shift uncertainty arrays.

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES

1. Compute the gain correction for an SGD event file using the MnK lines as the theoretical profile. The theoretical profile is convolved with a 200 eV Gaussian and gains are computed separately for each SIDE (the default values).
sgdgainfit infile=event_in.fits outevtsuffix=clean outfile=drift_out.fits

---
NAME sgdpipeline - SGD reprocessing tool

USAGE sgdpipeline indir outdir steminputs entry_stage exit_stage

DESCRIPTION
sgdpipeline duplicates most of the pipeline (not trend data) for the sgd. It allows the user to run all or part of the pipeline processing and to vary the calibration files and filtering (screening) criteria used. A number of other pipeline processing parameters can also be changed.

SGD Pipeline Stages

The sgdpipeline is divided into 3 stages:

Calibration
Data screening
Product creation

Each stage may be run singly or in combination with the preceding stages. This is controlled by the entry_stage and exit_stage parameters.

SGD Stage 1 consists of the following steps:
Run hxisgdsff
Run hxisgdpha
Run sgdevtid

The data screening (Stage 2) is identical to that in the production pipeline, when default parameters are used. For details on the default screening applied to the SGD events (respectively), see:
ahfilter - Create the EHK from attitude & orbital data and create the MKF from housekeeping data based on the CALDB mkfconf file
ahgtigen - Create the GTI from the EHK and MKF parameters based on CALDB selection file
ahscreen - Screen the data based on GTI and CALDB selection file
Default GTI used for screening data are:
GTIPOINT
GTITEL
GTIEHK
GTIMKF

The cleaning process occurs twice for SGD. Once to produce cleaned event files, the second to produce pseudo event files.

The product creation (Stage 3) is identical to that in the production pipeline, when default parameters are used. Extractor is run to create a lightcurve and spectra for each cleaned event file.

INPUT
The input to sgdpipeline is specified using (at minimum) the indir parameter. This should be specified as the sgd event_uf level sequence directory, e.g.:
sgdpipeline indir=/path/to/100039010/sgd/event_uf
Paths to specific sgd housekeeping and satellite data files can be specified using the attitude, housekeeping, extended_housekeeping, makefilter, orbit and obsgti parameters. The attitude, orbit and sgd housekeeping files are required for stage 1 calibration.

OUTPUT
Filenames, etc.

The number of output files depends on both pipeline processing stage(s) and the options selected. All output files are written to the directory specified by the outdir parameter. The archive directory structure is NOT reproduced (i.e. all output files is in a single directory).

The names of files produced are the same as those found in the HEASARC archive. However the usual "ahXXXXXXXXX" prefix, where "XXXXXXXXX" is the sequence number, can be replaced by a character string set by the stemoutputs parameter. This defaults to the value set by the steminputs parameter.

PARAMETERS
indir [string]
sgdpipeline:
Input directory

outdir [string]
sgdpipeline:
Output directory

steminputs [string]
sgdpipeline:
stem inputs

stemoutputs [string]
sgdpipeline:
stem outputs

instrument [string]
sgdpipeline:
Instrument (SGD,SGD1,SGD2)

entry_stage [integer]
sgdpipeline:
Entry stage

(exit_stage = 2) [integer]
sgdpipeline:
Exit stage

(sgd_start = 0.0) [real]
sgdpipeline:
SGD CALDB start time

verify_input [boolean]
sgdpipeline:
Verify with ftverify (yes, no)

(sgd_mkflabel = SGDSFFA1#) [string]
Label to use for SGD MKF GTI creation. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(sgd_ehklabel = SGDSFFA1#) [string]
Label to use for SGD EHK GTI creation. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(sgd_evtlabel = SGDSFFA1#) [string]
Label to use for SGD event screening. The hash is replaced by a respective Compton camera: CC1, CC2 or CC3
For pseudo events "PSE" is appended to the end of the label

(ra = -999.99999) [real]
The Right Ascension in decimal degrees of the point to appear in the center of SKY coordinate images. If ra is within the range 0-360 deg., then the value of the parameter is used. If ra is outside this range, then the ra value is read from the event header of the input file, searching first for the RA_NOM keyword and if found using its value, or if not found, then searching for the RA_PNT keyword and using its value. If neither keyword is found, then coordevt exits with an error. The default value for ra triggers the keyword look up.

(dec = -999.99999) [real]
The declination in decimal degrees of the point to appear in the center of SKY coordinate images. If dec is within the range -90 -- +90 deg., then the value of the parameter is used. If dec is outside this range, then the dec value is read from the event header of the input

file, searching first for the DEC_NOM keyword and if found using its value, or if not found, then searching for the DEC_PNT keyword and using its value. If neither keyword is found, then coordevt exits with an error. The default value for dec triggers the keyword look up.

(roll = 0.0) [real]
The roll angle about the center of the SKY coordinate system in decimal degrees. The roll angle is the angle measured counterclockwise from Celestial North to the positive SKY Y axis.

(sg1_optfocx = -999.99999) [real]
SGD1 optical focx coordinate

(sg1_optfocy = -999.99999) [real]
SGD1 optical focy coordinate

(sg1_optskyx = -999.99999) [real]
SGD1 optical detx coordinate

(sg1_optskyy = -999.99999) [real]
SGD1 optical dety coordinate

(sg2_optfocx = -999.99999) [real]
SGD2 optical focx coordinate

(sg2_optfocy = -999.99999) [real]
SGD2 optical focy coordinate

(sg2_optskyx = -999.99999) [real]
SGD2 optical detx coordinate

(sg2_optskyy = -999.99999) [real]
SGD2 optical dety coordinate

(sg1_ra_pnt = -999.99999) [real]
RA of SGD1 pointing [deg]

(sg1_dec_pnt = -999.99999) [real]
DEC of SGD1 pointing [deg]

(sg2_ra_pnt = -999.99999) [real]
RA of SGD2 pointing [deg]

(sg2_dec_pnt = -999.99999) [real]
DEC of SGD2 pointing [deg]

(extended_housekeeping = ah1001.ehk) [string]
ahgtigen:
Extended housekeeping file

(makefilter = ah1001.mkf) [string]
ahgtigen:
Makefilter file

(obsgti = ah1001_gen.gti) [string]
ahscreen:
Observation GTI file

(remapfile = CALDB) [filename]
hxisgdsff/sgdevtid:
File describing the remapping of ASIC_ID and READOUT_ID to sequential numbers in the columns ASIC_ID_RMAP and READOUT_ID_RMAP. Specify CALDB to retrieve the file from the calibration database.

(gainfile = CALDB) [filename]

hxisgdpha:
FITS table describing the gain calibration of PHA for the HXI or SGD camera. The domain of PHA values is divided into intervals, and a set of cubic polynomial coefficients is given for each interval. Specifying CALDB causes this file to be retrieved from the calibration database.

(badpixfile = CALDB) [filename]
hxisgdpha/sgdevtid:
FITS table giving the active and bad channels (pixels) of the HXI or SGD camera. Specifying CALDB causes this file to be retrieved from the calibration database.

(fluorefile = CALDB) [filename]
sgdevtid:
Calibration file containing the energy ranges of fluorescence electrons in CdTe and Si. Default: CALDB.

(probseqfile = CALDB) [filename]
sgdevtid:
Calibration file assigning the probability of each possible sequence of interactions.

(probfovfile = CALDB) [filename]
sgdevtid:
Calibration file containing the probability for photons to be in the SGD camera field of view, based on incident angle.

(leapsecfile = REFDATA) [filename]
ahgtigen/ahscreen:
Input leap second file (or CALDB, [REFDATA])

(selectfile = CALDB) [filename]
ahgtigen/ahscreen:
Input file with the selection expressions

(outnsubcol = no) [boolean]
hxisgdpha:
Output the PHA_NSUB column (yes/no). This column is the PHA minus the ASIC common mode noise (ASIC_CMN column in the SFF).

(datamode = NONE) [string]
hxisgdpha:
Substitute DATAMODE in place of event value (or NONE)

(rejectbgo = no) [boolean]
sgdevtid:
Reject events for which BGO trigger occurred (yes/[no]).

(skipreco = no) [boolean]
sgdevtid:
If yes, READALL and CALMODE occurrences are not reconstructed. Default: NO.

(outtracefile = NONE) [filename]
sgdevtid:
Output reconstruction tracing file (or NONE)

(numsignal = 48) [integer]
sgdevtid:
Maximum number of signals to analyze. Reconstruction is not attempted on occurrences with more than this number of signals.

(d10 = 3.2) [real]
sgdevtid:
[mm] Orthogonal distance between two adjacent pixels in the same detector layer; used in identifying charge sharing events.

(d1a1a = 5.0) [real]
sgdevtid:

[mm] Diagonal distance between two adjacent pixels in same detector layer; used in identifying possible CdTe-CdTe fluorescence signals in a single layer.

(d1a1b = 5.0) [real]
sgdevtid:
[mm] Distance between two CdTe layers; used in identifying possible CdTe-CdTe fluorescence signals in different layers.

(d1a2 = 14.0) [real]
sgdevtid:
[mm] Distance between CdTe and Si layers; used in identifying possible Si-CdTe fluorescence signals.

(d1a3 = 5.0) [real]
sgdevtid:
[mm] Distance between Si and Si layers; used in identifying possible electron scattering signals.

(a = 3.0) [real]
sgdevtid:
Acceptance tolerance for F test in Step 2

(b = 3.0) [real]
sgdevtid:
Acceptance tolerance for G test in Step 2

(probaccept2 = 0.1) [real]
sgdevtid:
Probability threshold for M=2 acceptance in Step 3

(probaccept3 = 0.1) [real]
sgdevtid:
Probability threshold for M=3 acceptance in Step 3

(probaccept4 = 0.1) [real]
sgdevtid:
Probability threshold for M=4 acceptance in Step 3

(distz = 1000000.0) [real]
sgdevtid:
Very large distance of target object in Step 4 [mm]. The Z-direction is toward the sky, so this distance represents the distance to an object far away from the detector.

(paraoffset0 = 1.6) [real]
sgdevtid:
Offset parameter combined with G[k,0] for the figure-of-merit (FOM) discrimination in Step 4 above.

(paraoffset1 = 1.0) [real]
sgdevtid:
Offset parameter combined with G[k,1] for the figure-of-merit (FOM) discrimination in Step 4 above.

(paraoffset2 = 1.0) [real]
sgdevtid:
Offset parameter combined with G[k,2] for the figure-of-merit (FOM) discrimination in Step 4 above.

(weight0 = 1.0) [real]
sgdevtid:
Parameter used in weighting G[k,0] for the figure-of-merit (FOM) discrimination in Step 4 above.

(weight1 = 0.0) [real]
sgdevtid:
Parameter used in weighting G[k,1] for the figure-of-merit (FOM) discrimination in Step 4 above.

(weight2 = 0.0) [real]
sgdevtid:

Parameter used in weighting G[k,2] for the figure-of-merit (FOM) discrimination in Step 4 above.

(weight3 = 0.0) [real]
sgdevtid:
Parameter used in weighting Prob[k] for the figure-of-merit (FOM) discrimination in Step 4 above.

(delgmethod = ANALYTIC) [string]
sgdevtid:
Method to be used to calculate the error in cos(theta_G) in the test of consistency between geometric and kinematic scattering angles. The possible values are "analytic" and "corner."

(occurrenceid = -1) [integer]
Occurrence to process (if >0)

(randomize = yes) [boolean]
Allow randomization ([yes]/no)

(seed = 0) [integer]
Random number generator seed (0=use system time)

(stemreport =) [string]
File stem for log and temporary files. If the parameter is not set the script automatically sets the stem to "sgdpipeline_YYYYMMDDTHHMMSS_" and appends log file and temp file names as needed. Intended to be set by ahpipeline.

(numerrs = 0) [string]
Number of errors from sgdpipeline (output)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

THE FOLLOWING PARAMETERS ARE SET BY THE SCRIPT INTERNALLY:
HXISGDSFF:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization. Input First FITS File (FFF) from HXI or SGD.

outfile [filename]
Set to the same name as infile, but in the outdir directory. Output Second FITS File (SFF). May be the same or a different file from infile. If outfile is the same file as infile, then clobbering must be enabled or the tool exits with an error. Clobbering is enabled by specifying clobber=yes (see below) or by prepending an exclamation point (!) to the output file name. Either of these methods causes the file conversion to be done in-place by modifying the input file.

HXISGDPHA:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff.
Input Second FITS File (SFF) from HXI or SGD.

outfile [filename]
Set to the same name as infile, but in the outdir directory. Output Second FITS File (SFF). If it is the same as infile, then specifying clobber=yes (see below) causes the file conversion to be done in-place by modifying the input file. Prepending an exclamation point (!) to the output file name has the same effect.

HXISGDEXPAND:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff and hxisgdpha. Input file of event data from an HXI instrument. This is the Second FITS File (SFF) after PHA calibration by the tool hxisgdpha.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "_uf.evt" to "exp_ufa.evt". Output event file containing results of event reconstruction.

HXIEVTID:
infile [filename]
Set based on the unfiltered event files found in indir directory during initialization and processed through hxisgdsff and hxisgdpha. Input file of event data from an HXI instrument. This is the Second FITS File (SFF) after PHA calibration by the tool hxisgdpha.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "_uf.evt" to "rec_ufa.evt". Output event file containing results of event reconstruction.

AHGTIGEN:
mkffile [filename]
Set to the makefiler file parameter the first time ahgtigen runs.
Set to the extended_housekeeping file parameter the second time ahgtigen runs (unless the files being generated are pseudo event files).

outfile [filename]
Set to ${sgdfile_out} . gtimkf, when the input file is set to the makefiler file parameter.
Set to ${sgdfile_out} . gtiehk, when the input file is set to the extended_housekeeping file parameter.

(instrume = NONE) [string]
(SGD1 or SGD2) Read from the header of the file being processed.

(label = NONE) [string]
("SGDSFFA1" . ${detnam}) or ("SGDSFFA1" . ${detnam} . "PSE") depending on if the files being generated are "clean" or "pseudo" respectively.

AHSCREEN:
infile [string]
Set systematically to the elements in an array which is constructed from any reconstructed unfiltered event files produced in the calibration stage. If the calibration stage is being skipped, this array is filled from searching the input directory instead.

outfile [filename]
Set to the same name as infile, but in the outdir directory and changing the extension from "rec_ufa.evt" to "rec_cl.evt".

(gtifile) [string]
Set as an array of gtifiles with corresponding extensions; pointing, tel, mkf, and (where applicable) ehk.

(upkeyword = YES) [boolean]
Update timing keywords from input file(s) in output file Hardcoded as yes

(label = NONE) [string]
("SGDSFFA1" . ${detnam}) or ("SGDSFFA1" . ${detnam} . "PSE") depending on if the files being generated are "clean" or "pseudo" respectively.

EXAMPLES
1. The following command recalibrates (stage 1) and re-screen (stage 2) all SGD data for sequence 100039010 that currently resides in the directory /data/100039010/sgd/event_uf, and the output is stored in a directory called /data/100039010_reproc/:

```
 sgdpipeline indir=/data/100039010/sgd/event_uf outdir=/data/100039010_reproc \
entry_stage=1 exit_stage=2 steminputs=ah100039010  obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti \
makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf  \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk
```

2. The following command re-screens (stage 2 only) SGD data for the same data set as in the previous example.

```
 sgdpipeline indir=/data/100039010/sgd/event_uf outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 \
steminputs=ah100039010 obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti \
makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk
```

3. The following command creates products (stage 3 only) SGD data for a calibrated data set:

```
 sgdpipeline indir=/data/100039010/sgd/event_cl outdir=/data/100039010_reproc entry_stage=3 exit_stage=3 steminputs=ah100039010
```

NOTES
None, but see help for individual parameters above.

NAME

skyback -- simulates the total, broadband, discrete and diffuse high-energy astrophysical background

USAGE skyback outfileroot ra dec radius emin emax de fluxsens fluxmin fluxmax bandpasslo bandpasshi slopebright1 slopefaint1 fluxbreak norm1 spectype1

DESCRIPTION

Skyback simulates the discrete and diffuse high-energy astrophysical background over a large energy range. Skyback currently includes the following three basic background components, any or all of which may be included.

(1) Galactic Halo and Local Hot Bubble (LHB) components are derived (following the QUICKSIM approach) from ROSAT all-sky survey (RASS) surface brightness maps (Snowden et al., ApJ, 485, 125) to create a model background spectrum appropriate for the part of the sky being observed. The model consists of an absorbed thermal plasma at T(halo)~1.0e6.6 K representing the excess over the extragalactic component seen in the RASS hard bands, plus an unabsorbed thermal plasma at T(LHB)~1.0e6 K representing the additional excess in the soft band. To generate this component, the skyback parameter flaggal must be set to "yes" (its default setting).

(2) Solar Wind Charge Exchange (SWCX) emission may be modeled as either a pure emission-line spectrum, or a continuous spectrum based on DXL sounding rocket measurements of the local galaxy diffuse X-ray emission (Galeazzi et al. 2014, Nature, 5121,171). The parameters that control the SWCX background component are: (a) flagswcx which must be set to yes, (b) swcxOVII and/or (c) swcxcont parameters. The swcxOVII parameter represents the total flux (in LU; 1 LU == 1 photon/s/cm^2/str) in the brightest (0.57 keV OVII) feature, and is used as follows. The flux is distributed among the forbidden, resonance, and intercombination OVII lines with the following scaling. F(OVII forbidden 0.5609 keV) = 2/3*F(0.57 keV OVII), F(OVII resonance 0.5740 keV) = 1/6*F(0.57 keV OVII), F(OVII intercombination 0.5685 keV) = 1/6*F(0.57 keV OVII). In addition the OVIII lines are added by scaling this input flux with the following factors: F(OVIII Lalpha 0.6536 keV) = 0.25*F(0.57 keV OVII), F(OVIII Kbeta 0.6657 keV) = 0.083*F(OVIII Lalpha 0.6536 keV). The swcxcont parameter represents the total flux in the 0.2-2 keV band of the continuous SWCX emission, modeled as 0.102 keV thermal (currently, Raymond-Smith) emission, in erg/cm^2/s/arcmin^2. The norm based on DXL observations corresponds to the default value ~3.17e-16 erg/cm^2/s/arcmin^2. If flagswcx is set to no, swcxOVII and swcxcont parameters are ignored.

(3) The third background component is modeled (based on the XIMAGE simulator) as a two-subcomponent logN-logS distribution of point sources, and is calculated if the parameter flaglogns is to yes (its default setting). The first logN-logS is a broken power-law designed to allow the inclusion of an extragalactic point-source component. The bright-end slope is determined by the parameter slopebright1, the faint-end slope by slopefaint1, the flux where the slope changes from slopebright1 to slopefaint1 is set by the parameter fluxbreak, and the normalization is set by the parameter norm1 in units of sources per square degree. The fluxes distributed according to the logN-logS distribution are defined over the energy range given by the parameters bandpasslo and bandpasshi (in keV). The parameters fluxmin and fluxmax are the minimum and maximum fluxes, in this bandpass, of the logN-logS distributions. The parameter fluxsens is the flux threshold defined such that sources fainter than fluxsen are treated as "undetected" and their emission subsumed into the diffuse background based on their individual spectra (see below). However, if fluxsen=0, the threshold is defined by the optional parameters ctstoflux (flux per unit count rate), sigtonoise (signal-to-noise ratio), and exposure (in seconds) according to ctstoflux*sigtonoise*sigtonoise/exposure (e.g., a value of cts_flux of 1.0e-11 corresponds to a threshold of 2.5e-15 erg/cm^2/s for logns_snr=5 and exposure=100 ks). The second logN-logS is a single power-law that may be used to create, for example, a Galactic or an additional extragalactic point source background. The optional parameters slope2 and norm2 determine its logN-logS slope, and normalization in sources per square degree. The optional parameters slope2 and norm2 determine its logN-logS slope, and normalization in sources per square degree. By default, norm2=0. The two logN-logS have several parameters in common -- namely bandpasslo, bandpasshi, fluxmin, fluxmax, fluxsen, ctstoflux, sigtonoise, and exposure.

Spectra for the point sources drawn from the broken power-law logN-logS background are assigned based on the parameter spectype1. If spectype1=0, the sources are assigned heasim-supported spectral models. In this case the model is specified by the skyback parameter specmod1: plaw (power-law), bbod (blackbod), brem (thermal bremsstrahlung), rs (Raymond-Smith thermal plasma), or mono (mono-energetic); and by the value of the parameter specpar1 (index for power-law, temperature in keV for blackbody/bremsstrahlung/Raymond-Smith, or line energy in keV for mono-energetic). The foreground column density is given by the parameter nhmod1. If spectype1=2, the background is given by the "torus" model based on the approach of Ueda et al. 2014 (ApJ, 786, 104). Each source spectrum is determined by one of the torus models from Brightman & Nandra 2011 (MNRAS, 413,1206; BN) with a distinct intrinsic column density and opening angle, and with a randomly selected inclination angle. The opening angle distribution is determined by fractions in five bins (75) specified by the parameters fpar0, fpar1, fpar2, fpar3, and fpar4. The intrinsic column density distribution is determined by fractions in six logarithmic NH bins (25) specified by the parameters fabs0, fabs1, fabs2, fabs3, fabs4, and fabs5. Redshifts are drawn from a distribution assumed to be a simple linear ramp up to z=1, followed by an exponential decline. If spectype1=1, the background is given by the "multi" model based on the approach of Gilli, Comastri, & Hasinger, G. 2007 (A&A, 463, 792) where each point source is an absorbed power-law with index and NH drawn from two distributions. The power-law index distribution is determined by fractions in five index bins (1.5-1.7, 1.7-1.9, 1.9-2.1, 2.1-2.3, 2.3-2.5) specified by fpar0, fpar1, fpar2, fpar3, and fpar4. Again, the intrinsic column density distribution is determined by fractions in six logarithmic NH bins (25) specified by

fabs0, fabs1, fabs2, fabs3, fabs4, and fabs5; and redshifts assigned as described above. For "multi" models, in the highest NH-bin the power-law is replaced by a reflection model approximated by a BN torus model with the largest available opening angle, inclination of 0 degrees, the assigned power-law, and the assigned (>1.0e25 1/cm^2) NH - with the power-law subtracted off. In the next-to-highest NH-bin the absorbed power-law is supplemented with this same reflection model, scaled down by 0.37 (Gilli et al. 2007).

Spectra for the point sources drawn from the single power-law logN-logS background are assigned in an identical way to those of the broken power-law logN-logS background if samespec=YES (i.e., the lists are merged before the spectra are assigned). If samespec=NO, the sources are assigned heasim-supported spectral models. In this case the model is specified by parameter specmod2: plaw (power-law), bbod (blackbod), brem (thermal bremsstrahlung), rs (Raymond-Smith thermal plasma), or mono (mono-energetic). The value of the spectral parameter for this model is specified by specpar2, and the foreground column density by the parameter nhmod2, as described above for specmod1 and nhmod1.

If flaglogns = yes, slopebright, slopefaint1, fluxbreak, norm1, bandpasslo, bandpasshi, fluxmin, fluxmax, fluxsens, and spectype1 are required and must be set. If flaglogns = no these are ignored, as are all of the other, optional, parameters associated with the logN-logS component such as those associated with the single power-law logN-logs or with the details of the spectral model.

The skyback output may include the following files. (1) A catalog of resolved point sources that includes source positions with spectra specifications. The format follows the heasim source definition file for point sources. The name of the output is constructed by appending the value of the parameter outfileroot to "pscat". (2) A file that supplements the point source catalog contains values of the redshift and intrinsic absorption. The name of the output is constructed by appending the value of the parameter outfileroot to "pszcat". (3) A "catalog" with a single line representing the diffuse emission. The format follows the heasim source definition file format for an extended source with a flat distribution. The flat distribution is centered at the position given by the parameters ra and dec, and covers a circular region extending out to parameter radius in arcmin. The name of the output is constructed by appending the value of the parameter outfileroot to "difcat". (4) The total diffuse background spectrum corresponding to (3) in the form of a heasim user input spectrum ASCII file (flux in photons/cm^2/s/channel versus energy in keV). The diffuse spectrum is calculated on a user-specified energy grid in keV ranging from the parameter emin to the parameter emax with a constant grid spacing given by the parameter de. For example, one might select emin=0.1, emax=16.0, and de=0.0005 for the Hitomi SXS and emin=0.1, emax=120.0, and de=0.025 for the Hitomi HXI. The name of the output ASCII file is constructed by appending the value of the parameter outfileroot to "difspec." (5) An XSPEC table model (FITS) file is also output, with a name constructed by appending the value of the parameter outfileroot to "difspec." This may be directly examined, e.g., via the command "model atable{outfileroot_difspec.fits}" within XSPEC.

PARAMETERS
outfileroot [string]
The root name used to construct the filenames of the skyback outputs. The outputs are: pscat_outfileroot.txt, pszcat_outfileroot.txt, difcat_outfileroot.txt, difspec_outfileroot.dat, difspec_outfileroot.fits. Required.

ra [double]
The right ascension of the region center to be populated with background (discrete and diffuse) sources. This parameter is given in decimal degrees, J2000 epoch. Required.

dec [double]
The declination of the region center to be populated with background (discrete and diffuse) sources. This parameter is given in decimal degrees, J2000 epoch. Required.

radius [double]
Radius, in arcminutes, of the circular region to be populated with background (discrete and diffuse) sources. Required.

emin[double]
Energy, in keV, of the lower limit of the energy grid for the computed background spectrum. Required.

emax [double]
Energy, in keV, of the upper limit of the energy grid for the computed background spectrum. Required.

de [double]
Constant energy spacing, in keV, of the energy range for the background spectrum. Required.

(flaglogns = yes) [boolean]
If yes (the default), the background component originating from point sources drawn from a logN-logS distribution is computed. Optional [yes]/no).

fluxsens [double]

Flux for the point source sensitivity limit used for both logN-logS distributions. Sources with flux above this threshold are treated as discrete point sources, sources below are treated as unresolved and subsumed into the diffuse background. Required if flaglogns=yes, but may be set to 0, in which case the threshold is determined by the exposure, sigtonoise, and ctstoflux parameters. Ignored if flaglogns=no.

fluxmin [double]
Flux lower limit, in erg/cm^2/s, used for both logN-logS distributions. Required if flaglogns=yes, otherwise ignored.

fluxmax [double]
Flux upper limit, in erg/cm^2/s/arcmin^2, used for both logN-logS distributions. Required if flaglogns=yes, otherwise ignored.

bandpasslo [double]
Bandpass lower limit, in keV, over which both logN-logS distributions are defined. Fluxes output in the "pscat_fileroot" file refer to the energy range defined by bandpasslo and bandpasshi. Required if flaglogns=yes, otherwise ignored.

bandpasshi [double]
Bandpass upper limit, in keV, over which both logN-logS distributions are defined. Fluxes output in the "pscat_fileroot" file refer to the energy range defined by bandpasslo and bandpasshi. Required if flaglogns=yes, otherwise ignored.

slopebright1 [double]
Bright-end slope of the first (broken power-law) logN-logS distribution. Required if flaglogns=yes, otherwise ignored.

slopefaint1 [double]
Faint-end slope of the first (broken power-law) logN-logS distribution. Required if flaglogns=yes, otherwise ignored.

fluxbreak [double]
Flux level, in erg/cm^2/s/arcmin^2, where the first (broken power-law) logN-logS distribution changes slope, i.e. slope=slopebright1 (slopefaint1) above (below) fluxbreak. Required if flaglogns=yes, otherwise ignored.

norm1 [double]
Normalization, in sources/square-degree, of the first (broken power-law) logN-logS distribution. Required if flaglogns=yes, otherwise ignored.

spectype1 [integer]
Type of spectral model assigned to the point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The values for spectype1 are 0, 1, and 2 with the following meanings. If spectype1=0, all sources have identical spectral models as specified in the parameters specmod1, specpar1, and nhmod1. If spectype1=1 ("multi"), spectra are absorbed power-laws with column densities and slopes drawn from the distributions specified in the parameters fabs0-5 and fpar0-4, respectively, with a special treatment in the mildly and fully Compton thick regimes. If spectype1=2 ("torus"), spectra are Brightman & Nandra torus models with column densities and opening angles drawn from the distributions specified by fabs0-5 and fpar0-4, respectively, and with randomly selected inclination angles. Required if flaglogns=yes, otherwise ignored.

(specmod1 = pow) [string]
Available spectral models that are assigned to the point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. These are: plaw (power-law), bbod (blackbody), brem (thermal bremsstrahlung), rs (Raymond-Smith thermal plasma), or mono (mono-energetic). Optional, ignored if flaglogns=no.

(specpar1 = 1.9) [double]
Parameter value of the spectral model specified in specmod1. The possible parameters are: index (for power-law model), temperature in keV (for blackbody, bremsstrahlung, Raymond-Smith models), or line energy in keV (for mono-energetic model). Optional, ignored if flaglogns=no.

(nhmod1 = 0.0) [double]
Value for the column density in 1/cm^2 used with the spectral model specified in specmod1. Optional, ignored if flaglogns=no.

(fabs0 = 0.24) [double]
Fraction of sources which have assigned NH<1.0e21 1/cm^2 and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The value for the fraction ranges between 0-1. Optional, ignored if flaglogns=no.

(fabs1 = 0.24) [double]

Fraction of sources which have assigned NH between 1.0e21 and 1.0e22 1/cm^2 and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The fraction values range between 0-1. Optional, ignored if flaglogns=no.

(fabs2 = 0.24) [double]
Fraction of sources which have assigned NH between 1.0e22 and 1.0e23 1/cm^2 and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The fraction values range between 0-1. Optional, ignored if flaglogns=no.

(fabs3 = 0.24) [double]
Fraction of sources which have assigned NH between 1.0e23 and 1.0e24 1/cm^2 and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The fraction values range between 0-1. Optional, ignored if flaglogns=no.

(fabs4 = 0.03) [double]
Fraction of sources which have assigned NH between 1.0e24 and 1.0e25 1/cm^2 ("mildly Compton thick" regime) and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The fraction values range between 0-1. Optional, ignored if flaglogns=no.

(fabs5 = 0.01) [double]
Fraction of sources which have assigned NH > 1.0e25 1/cm^2 ("Compton thick" regime) and are drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. This is applicable if spectype1 is either 1 or 2. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(fpar0 = 0.2) [double]
Fraction of sources which have assigned a power law index between 1.5 and 1.7 if spectype1=1 or torus opening angles < 30 degrees if spectype1=2. Applied to point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(fpar1 = 0.2) [double]
Fraction of sources which have assigned a power law index between 1.7 and 1.9 if spectype1=1 or torus opening angles between 30 and 45 degrees if spectype1=2. Applied to point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(fpar2 = 0.2) [double]
Fraction of sources which have assigned a power law index between 1.9 and 2.1 if spectype1=1 or torus opening angles between 45 and 60 degrees if spectype1=2. Applied to point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(fpar3 = 0.2) [double]
Fraction of sources which have assigned a power law index between 2.1 and 2.3 if spectype1=1 or torus opening angles between 60 and 75 degrees if spectype1=2. Applied to point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(fpar4 = 0.2) [double]
Fraction of sources which have assigned a power law index between 2.3 and 2.5 if spectype1=1 or torus opening angles between 75 and 90 degrees if spectype1=2. Applied to point sources drawn from the first logN-logS distribution (broken power-law) - as well as the second if samespec=YES. The fraction values ranges between 0-1. Optional, ignored if flaglogns=no.

(slope2 = 1.1) [double]
Slope of the second (single power-law) logN-logS distribution. Optional, ignored if flaglogns=no.

(norm2 = 0.0e0) [double]
Normalization, in sources/square-degree, of the second (single power-law) logN-logS distribution. Optional, ignored if flaglogns=no.

(samespec = yes) [boolean]
If samespec=YES, the spectral model for the point sources from the second logN-logS distribution (single power-law) is assigned in the same way as for the point sources from the first logN-logS distribution (broken power-law), i.e. identical spectra if spectype1=0 or spectra drawn from identical spectral parameter distributions if spectype=1 or 2. Optional, ignored if flaglogns=no ([yes]/no).

(specmod2 = pow) [string]

Available spectral models that are assigned to the point sources drawn from the second logN-logS distribution (broken power-law) if samespec=NO. These are: plaw(power-law), bbod (blackbody), brem (thermal bremsstrahlung), rs (Raymond-Smith thermal plasma), or mono (mono-energetic). Optional, ignored if flaglogns=no.

(specpar2 = 1.9) [double]
Parameter value of the spectral model specified in spec_mod2. The possible parameters are: index (for power-law model), temperature in keV (for blackbody, bremsstrahlung, Raymond-Smith models), or line energy in keV (for mono-energetic model). Optional, ignored if flaglogns=no.

(nhmod2 = 0.0) [double]
Value for the column density in 1/cm^2 used with the spectral model specified in spec_mod2. Optional, ignored if flaglogns=no.

(flaggal = yes) [boolean]
If yes, the diffuse background component originating from Galactic halo and Local Hot Bubble hot gas is computed. Optional ([yes]/no).

(flagswcx = no) [boolean]
If yes, the background component originating from Solar Wind Charge Exchange is computed. Optional (yes/[no]).

(swcxOVII = 0.0) [double]
Total flux, in units of LU (1 LU = erg/cm^2/s/arcmin^2), of the OVII line complex for the discrete SWCX component. Optional, ignored if flagswcx=no.

(swcxcont = 3.17e-16) [double]
Total flux, in erg/cm^2/s/arcmin^2, of the continuous SWCX component. Optional, ignored if flagswcx=no.

(flagdgrb = no) [boolean]
NOT IMPLEMENTED. If yes, the diffuse gamma-ray background component is computed. Optional(yes/[no]) .

(flaggrxe = no) [boolean]
NOT IMPLEMENTED. If yes, the Galactic Ridge background component is computed. Optional (yes/[no]).

(exposure = 10000.) [double]
Exposure time in seconds used to calculate the point source logN-logS flux threshold (in combination with sigtonoise and ctstoflux, i.e. as ctstoflux*sigtonoise*sigtonoise/exposure if an explicit value (fluxsens) is not input. The threshold determines which sources are treated as discrete background sources and which as contributing to the diffuse background emission. Optional.

(sigtonoise = 5.0) [double]
Minimum signal-to-noise ratio used (in combination with exposure and ctstoflux) to determine the flux threshold if fluxsen=0. Optional.

(ctstoflux = 1.0e-11) [double]
Count-rate to flux conversion used (in combination with sigtonoise and ctstoflux) to determine the flux threshold if fluxsen=0. Optional.

(seed = 1) [integer]
Value with which to seed the simulator's random number generator (RNG). If set to zero, the seed is ignored and the RNG is seeded from the system time. Otherwise it is used as given and consecutive applications with identical seed parameter yield identical output. Optional.

(clobber = yes) [boolean]
Overwrites the existing output file if set to yes ([yes]/no).

(debug = no) [boolean]
Diagnostic output is printed out on the screen if set to yes (yes/[no]).

(mode = ql) [string]
Mode to query the parameter file. Acceptable values include: "ql (query and learn/remember), "hl" (hidden and learn/remember), "q" (query but don't remember), "h" (hidden).

EXAMPLES
1. Create skyback point source, diffuse source, and auxiliary point source catalog output files pscat_heasim.txt, difcat_heasim.txt, and pzcat_heasim.txt for a background field centered at (ra,dec) = (150.0,50.0) and radial extent of 10 arcmin. Create a diffuse spectrum text file difspec_heasim.dat, and xspec table model file difspec_heasim.fits. The background components include the "galactic" and "logN-logS" components, where the latter consists of point sources drawn from a single broken-power law logN-logS distribution with default

parameters and identical index 1.9 power-law spectra. The output spectra are created on an energy grid spanning 0.1-12 keV with 1 eV grid spacing.
skyback outfileroot=heasim ra=150.0 dec=50.0 radius=10.0 emin=0.1 emax=12.0 de=0.001 flaglogns=yes fluxsens=1.0e-14 \
fluxmin=1.0e-16 fluxmax=5.0e-13 bandpasslo=0.5 bandpasshi=2.0 slopebright1=1.7 slopefaint1=0.9 \
fluxbreak=2.5e-14 norm1=8.0e3 spectype1=0

2. Create skyback point source, diffuse source, and auxiliary point source catalog output files pscat_cxb_hxi.txt, difcat_cxb_hxi.txt, and pszcat_cxb_hxi.txt for a background field centered at (ra,dec) = (151.8606,16.1085) and radial extent of 20 arcmin. Create a diffuse spectrum text file difspec_cxb_hxi.dat, and xspec table model file difspec_cxb_hxi.fits. The background components include the "galactic" and "logN-logS" components, where the latter consists of point sources drawn from a single broken-power law logN-logS distribution defined in the 2-10 keV bandpass with a bright-end slope of 1.7, faint-end slope of 0.9, flux where the slope breaks of 3.5e-14 erg/cm^2/s, and 19000 source/deg2 between 1.0e-16 and 7.0e-13 erg/cm^2/s. The spectra are "torus" models drawn from a distribution with probabilities of 0.2, 0.2, 0.2, 0.2, 0.1, and 0.1 for the 6 defined column density bins and equal probabilities in the 5 defined torus opening angle bins. The output spectra are created on an energy grid spanning 0.1-120 keV with 25 eV grid spacing.
skyback outfileroot=cxb_hxi ra=151.8606 dec=16.1085 radius=20.0 emin=0.1 emax=120.0 de=0.025 fluxsens=1.0e-14 \
fluxmin=1.0e-16 fluxmax=7.0e-13 bandpasslo=2.0 bandpasshi=10.0 slopebright1=1.7 slopefaint1=0.9 \
fluxbreak=3.5e-14 norm1=1.9e4 spectype1=2 fabs0 = 0.2 fabs1 = 0.2 fabs2 = 0.2 fabs3 = 0.2 fabs4 = 0.1 fabs5 = 0.1

NAME sxiflagpix -- Flag pixel STATUS for SXI event data

USAGE sxiflagpix infile outfile outbadimg hotpixfile flickpixfile

DESCRIPTION

'sxiflagpix' flags SXI events that may be unsuitable to use in data analysis, such as those that fall on defective or non-sensitive pixels; those that might have missing or compromised telemetry; and those for which processing has encountered certain errors. The STATUS column is updated, encoding the flagging condition for each event. This tool does not actually filter or remove events, however the STATUS column can be used for later filtering. An image in DET coordinates showing the location of unsuitable pixels is also output for later use in constructing an ARF or flat field image.

The tool uses several input files to assign STATUS: a list of bad or dead pixels or columns that don't change ("badpixfile"); a list of hot pixels, which can change on short timescales and are observation-specific ("hotpixfile"); a list of flickering pixels, which can be produced by the task searchflickpix ("flickpixfile"); and a mask file that encodes the location of CCD boundaries, segments, calibration source regions, and other regions that don't change ("maskfile"). The locations of the charge injection rows are read from the input event list header, as are settings for window mode and area discrimination.

In addition to an output event list with updated STATUS column, sxiflagpix optionally outputs a bad pixel list (specified by the parameter 'outbadpix') containing the events that have any STATUS flag set by the tool. Also output is an image (specified by the parameter 'outbadimg') in SXI DET coordinates containing a value for each pixel corresponding to its STATUS. The values in these files are, in order of increasing precedence:

0: a good pixel
1: a pixel inside the calibration source region
2: a bad pixel set by the parameter "bad_status"
-1: a pixel out of the CCD, window, or area discrimination region

For example, a pixel inside the calibration region but flagged as bad has value 2 in the output image. Flickering pixels are not included in this map because they can depend on time. Note that the output bad pixel list contains only those pixels for which events are found in the input file, while the output image contains all pixels.

Below is a table mapping flag bit position in the STATUS column to particular flagging conditions. If the flag is set to 1, that condition is flagged. Unless otherwise noted, the STATUS values apply to the central pixel of an event. Tools other than sxiflagpix that set STATUS values are noted in parentheses. "CI" indicates "charge injection."

flag	description	
1	All bad events set by "bad_status" parameter	
2	Inside the calibration source region	
3	Out of CCD	(Out of area)

```
4     Out of window
5     Out of area discrimination
-------------------------------------------------------------------------------
6     CI row                                    (Pixels)
7     Bad pixel from CALDB
8     Bad column from CALDB
9     Hot pixel from pre-pipeline
10    Flickering pixel
-------------------------------------------------------------------------------
11    CCD boundary                              (Boundaries)
12    Window boundary
13    Segment boundary
14    Area discrimination boundary
15    At least one 3x3 surrounding pixel has a bad status
-------------------------------------------------------------------------------
16    CI trailing row                           (Neighbors)
17    CI preceding row
18    Preceding/following of bad column
19    Neighbors of bad/hot pixel and bad column
20    Neighbors of flickering pixel
21    Neighbors of preceding/following of bad column
22    Neighbors of CCD/window boundary
23    Neighbors of segment boundary
-------------------------------------------------------------------------------
24    (sxiphas) 3x3 info is present but 5x5 is absent    (Others)
25    (sxiphas) 3x3 is absent
26    (sxipi - general) PHAS[0] < event threshold
27    (sxipi - vtevnodd) Video temperature is out of range
28    (sxipi - vtevnodd) Lack of video temp HK at time close to the event
29    (sxipi - chtrail/CTI) Correction value is negative
30    (sxipi - general) Null value by correction process
-------------------------------------------------------------------------------
31    1st trailing row of the CI rows           (Diagnostics)
32    1st preceding row of the CI rows
33    2nd trailing row of the CI rows
34    2nd preceding row of the CI rows
35    3rd trailing row of the CI rows
36    3rd preceding row of the CI rows
-------------------------------------------------------------------------------
37    Cosmic ray echo pixel                     (Cosmic Rays)
-------------------------------------------------------------------------------
38-48  Reserved
===============================================================================
```

The special STATUS flag 1 is set for any event that has a STATUS flag listed in the "bad_status" parameter. This is to enable easy filtering syntax later in the processing, since good events can be extracted with a simple filter expression.

Several STATUS conditions apply to events where neighboring pixels are flagged. Because an event is composed of a 3x3 pixel island, the quality of the region immediately surrounding the event center must be taken into account. For example, a bad pixel in the outer ring of a 3x3 island does not produce a valid pulse height, and this could result in a event with an otherwise unacceptable grade being accepted as a good event. The parameters 'npixnbr' and 'nboundnbr' can be used to set the maximum distance away from a bad pixel or boundary to flag further pixels and events. In addition, for more flexibility, sxiflagpix writes a new column PHAS_MASK to the output event list. This is a 9-element mask of the PHAS column, with each element indicating whether the corresponding pixel i in PHAS is good (PHAS_MASK[i]=0) or bad (PHAS_MASK[i]=1). PHAS_MASK can be used later in sxipi to determine how to grade the event and calculate the summed PHA, using the sxipi 'badpixopt' parameter. Since sxipi can overwrite the PHAS column, the ability to reset it to the original telemetry values is provided here with the 'copyphas' parameter. Similarly, 'resetflags' can be used to reset all the sxiflagpix STATUS flags and the elements of PHAS_MASK to zero. Using 'resetflags=yes' does not change the flags shown above that are set by sxipi or sxiphas.

Some pixels are affected by cosmic ray echo, in which a cosmic ray detected during an SXI dark frame can cause an echo in a different readout segment. This echo then causes those pixels to have a pulse height above the split threshold for the duration of the observation, leading to incorrect grades for events centered in neighboring pixels. Specifying 'echoflag=yes' causes 'sxiflagpix' to search for such

pixels in all SXI segments except the aimpoint segment (CCD_ID==1, SEGMENT==1), which is free of this effect. If at least 'echofrac' of the events containing a given pixel have a pulse height greater than or equal to 'echospth' in that pixel, then the pixel is considered a cosmic ray echo pixel. Any events centered 'echonbr' pixels from such pixels have STATUS flag 37 set, and if flag 37 is specified in 'bad_status', any pixels within 'echonbr' of that pixel will be marked as bad in the 'outbadimg'. For SXI, 'echonbr=2' is the recommended value, which will flag a 5x5 area of pixels around any cosmic ray echo pixel. Only pixels that have data from at least 'echomin' events are considered. An image of the echo fraction can be written out as an extension to 'outbadimg' if 'echomap=yes', and used for diagnostic purposes.

Charge trailed or leaked from charge injection rows can produce spurious events in several rows immediately surrounding the injected rows, manifesting as noise at soft energies. The parameters 'citrailnbr' and 'ciprenbr' can be used to tune a compromise between lower noise and higher efficiency, since flagging additional rows reduces the effective area.

Some examples of how to use the STATUS flags in filtering and extracting events are shown below in the "Examples" section. Also see the help for 'calc_express' for proper filtering syntax.

PARAMETERS

infile [filename]
Name of input SXI FITS event list.

outfile [filename]
Name of output SXI FITS event list.

(outbadpix) [filename]
Name of output list of events with bad pixels. If set to NONE, the task does not output this file.

outbadimg [filename]
Name of output image containing the bad pixel status. If set to NONE, the task does not output this file. If 'echoflag=yes' and 'echomap=yes', then this output file also contain a map of the fraction of cosmic ray echo pixels in the first image extension.

(badpixfile = CALDB) [filename]
Name of input bad pixel list. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information.

hotpixfile [filename]
Name of observation-specific input hot pixel list. If set to NONE, the task does not use this information.

flickpixfile [filename]
Name of observation-specific input flickering pixel list. If set to NONE, the task does not use this information.

(maskfile = CALDB) [filename]
Name of input mask file for the calibration sources and boundary regions. If set to CALDB, the file is read from the calibration database.

(npixnbr = 1) [integer]
Distance in pixels from a hot pixel, bad pixel, or bad column to flag a neighbor pixel.

(nboundnbr = 1) [integer]
Distance in pixels from a boundary to flag a neighbor pixel.

(citrailnbr = 2) [integer]
Distance in pixels trailing a charge injection (CI) row to flag a neighbor pixel.

(ciprenbr = 1) [integer]
Distance in pixels preceding a charge injection (CI) row to flag a neighbor pixel.

(echoflag = yes) [boolean]
If set to yes, cosmic ray echo pixels are flagged ([yes]/no).

(echonbr = 2) [integer]
Distance in pixels from a cosmic ray echo pixel to flag a neighbor pixel. If 'echoflag=no', this parameter is ignored.

(echomin = 6) [integer]

Minimum number of events for the cosmic ray echo fraction calculation. If 'echoflag=no', this parameter is ignored.

(echospth = 15) [integer]
Split threshold for cosmic ray echo fraction calculation. If 'echoflag=no', this parameter is ignored.

(echofrac = 0.7) [float]
Minimum fraction of hits defining a cosmic ray echo pixel. For any pixel contained in at least 'echomin' events, if at least 'echofrac' of those events have a pulse height above 'echospth', then the pixel is considered a cosmic ray echo pixel. If 'echoflag=no', this parameter is ignored.

(echomap = no) [boolean]
If set to yes, the cosmic ray echo pixel fraction map is output (yes/[no]). This is an image in DET coordinates that shows the fraction of pixels with pulse height above 'echospth' and can be used for diagnostic purposes. The image is saved as the first extension in 'outbadimg'. If 'outbadimg=NONE' or 'echoflag=no', this parameter is ignored.

(gtifile = NONE) [filename]
Name of input GTI file for cosmic ray echo fraction calculation. If a file is specified, only events within the GTI are used to determine which pixels are affected by cosmic ray echo. If set to NONE, all events in 'infile' are used.

(bad_status = "3:12,16:20,25:28,30,37") [string]
List of status flags which are considered "bad". These is used to set STATUS flag 1 and in the output bad pixel image. Colons can be used to specify a range (e.g., 3:5,7 = 3,4,5,7).

(copyphas = yes) [boolean]
If set to yes, the task copies the original PHAS column before processing ([yes]/no). This overwrites the PHAS column with the contents of PHAS_INNER3X3, which is the original telemetered pixel pulse heights of the 3x3 event island.

(resetflags = yes) [boolean]
If set to yes, the task resets all sxiflagpix STATUS flags (i.e., flags 1-23) ([yes]/no).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Run sxiflagpix with default parameters.
sxiflagpix infile="input.evt" outfile="output.evt" outbadimg="outbadimg.fits" hotpixfile="hotpix.fits" flickpixfile="flickpix.fits"

2. Run sxiflagpix without bad pixel file and flickering pixel file.
sxiflagpix infile="input.evt" outfile="output.evt" outbadimg="outbadimg.fits" hotpixfile="hotpix.fits" flickpixfile=NONE \
badpixfile=NONE

3. Run sxiflagpix with more restrictive flagging of rows neighboring the charge injection rows.
sxiflagpix infile="input.evt" outfile="output.evt" outbadimg="outbadimg.fits" hotpixfile="hotpix.fits" flickpixfile="flickpix.fits" \
citrailnbr=4 ciprenbr=2

4. Run sxiflagpix with non-default bad_status setting.
sxiflagpix infile="input.evt" outfile="output.evt" outbadimg="outbadimg.fits" hotpixfile="hotpix.fits" flickpixfile="flickpix.fits" \
bad_status="3:9,11:12,16"

(After running sxiflagpix) extract only those events from the output, which are "good", i.e. which have STATUS flag 1 set to 0. Note the use of a vector element, which in this case is STATUS[1] to get flag 1. Also note the bit mask "b0". ftcopy "output.evt[events][STATUS[1]==b0]" output_filtered.evt

(After running sxiflagpix) extract good events (STATUS flag 1 = 0) only in the calibration source regions (STATUS flag 2 = 1). ftcopy "output.evt[events][STATUS[1]==b0 && STATUS[2]==b1]" output_filtered.evt Another way to do the same filtering, using the "bxxx" bit mask notation, which is explained fully under "calc_express". The wild card "x" indicates any value is allowed.
ftcopy \
"output.evt[events][STATUS==b01xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]" output_filtered.evt
---
NAME sxigainfit -- Calculate the SXI time-dependent energy gain corrections for events from comparison with known calibration lines

USAGE sxigainfit infile outevtsuffix outfile

DESCRIPTION
'sxigainfit' is a script that runs 'ahgainfit' on SXI event files and allows the user to select and reprocess the events input into 'ahgainfit'. The description of the task 'ahgainfit' and its parameters also used by 'sxigainfit' are described below. 'sxigainfit' works with a single event file or list of files.

The SXI has calibration sources that emit Mn K lines mounted so that they illuminate one corner of each CCD. 'sxigainfit' calculates the time dependent shift of the calibration source energy. 'sxigainfit' includes options to (a) create gti (see parameter 'rungtigen'), (b) screen the events ('runscreen'), (c) select the calibration source region ('runselcal'), and (d) recalculate the PI ('runsxipi') prior to running 'ahgainfit'.

(a) If 'rungtitgen=yes', 'sxigainfit' creates a GTI file. The GTI are created using the expression either specified in the parameter 'gtiexpr' and/or using the default expression label ('gtigenlabel' parameter) stored in a CALDB file ('selectfile' parameter). The expression label refers to entries listed in the input mkf file ('gtigen_infile' parameter). If the 'gtifile' parameter is set, the single GTI file or a list of GTI files specified is also used.
(b) If 'runscreen=yes', 'sxigainfit' screens the input event data with the expression specified in the parameter 'expr' and/or using the default expression label ('screenlabel' parameter) stored in the CALDB file specified by the 'selectfile' parameter.
(c) If 'runselcal=yes', a region selection is applied to include only events from the calibration source location.
(d) If 'runsxipi' is set to 'FULL' or 'PARTIAL', the PI is recalculated. If 'runsxipi=FULL', sxipi is run in its default mode. If 'runsxipi=PARTIAL', sxipi is run with no charge trail correction, no CTI correction, and does not enable the split threshold iteration. The 'sxipi' parameters 'hkfile', 'hkext', 'hkcolstem', 'hkvideoid', 'vtevnoddfile', 'chtrailfile', 'ctifile ', 'spthfile', 'gainfile', 'patternfile', 'randomize', and 'seed' may be set within 'sxigainfit' (see 'sxipi' for details). If 'runsxsipi=no', sxipi is not run and the existing PI is used.

'sxigainfit' creates two outputs. The first is a gain correction file identical in structure to that produced by 'ahgainfit'. The second is the screened event file created within 'sxigainfit'. The filename of the screened event file is the same as the input file with a suffix (see 'outevtsuffix' parameter) appended.

ahgainfit:

'ahgainfit' calculates time-dependent energy gain corrections by comparing the theoretical and observed energies of a calibration line or line complex. For each run of the task, only one line may be specified to calculate the gain correction (see parameter 'linetocorrect'). 'ahgainfit' is a general task that is used directly in the scripts 'sxigainfit', 'hxigainfit', 'sgdgainfit' for the SXI HXI, and SGD respectively. For the SXS, the 'sxsgain' tasks uses the same core fitting function of 'ahgainfit'.

'ahgainfit' takes as input an event file with time and energy columns, and requires that the events are time ordered. The task accumulates spectra from events that are consecutive in time, with energy centered on the calibration feature and compares each spectrum from the events with a theoretical model of the calibration feature profile. The calibration feature used by 'ahgainfit' is specified by the parameter 'linetocorrect' as a string, and the names and energies of the features are specified in a calibration file (parameter 'linefitfile'). The calibration feature may be composed of many atomic or nuclear line components that are listed in the calibration file.

The energy range for the spectra constructed from the event file as well as from the theoretical profile, may be specified in two different ways. (1) The default energy range for the spectra is the smallest and largest energies of the line components, and expanded with the 'extraspread' parameter, i.e. [E_min - extraspread : E_max + extraspread]. It is recommended to set 'extraspread' larger than the sum of the natural width of the calibration feature, the value of the 'broadening' parameter, and the magnitude of the expected energy shift. (2) Alternatively, the energy range may be specified by setting the 'startenergy' and 'stopenergy' parameters. If these parameters are non-negative 'ahgainfit' uses their values to accumulate the spectra, instead of the range derived using the 'extraspread' setting. The energy column used to accumulate the spectra is specified by the 'energycol' parameter. The energy column is expected to be in units of channel, where the eV per channel is set by the 'evchannel' parameter. The spectra are binned according to the 'binwidth' parameter, where the 'binwidth' is in units of 'energycol' channels. The theoretical profile is constructed on a mesh defined by this energy range and 'binwidth', where each calibration line is assumed to be Lorentzian. The profile may be convolved with a Gaussian having the FWHM given by the 'broadening' parameter.

The number of events in each spectrum is defined by the 'numevent' and 'minevent' parameters. The task accumulates spectra with a number of events between 'minevent' and 'numevent'. However, if a spectrum has fewer than 'minevent' points, then it is combined with the previous spectrum if possible. Therefore all spectra have a size between 'minevent' and ('numevent+minevent-1'). To avoid having spectra accumulated over large gaps in time, the group of points in the spectrum is truncated when the time interval between consecutive events is greater than the 'gapdt' parameter. Adjacent spectra in time may share a percentage of their points based on the 'grpoverlap' parameter that may vary between 0 and 100. If 'grpoverlap' is set to 0, the consecutive spectra share no points in common; if set to 100 they share all points in common but one.

The spectra events may be simultaneously collected based on the value of a column present in the event file, given by the 'splitcol' parameter. This option may be used, for example, to find the gain correction for each layer of the HXI detector ('splitcol=LAYER'). If a GTI file is specified by the 'gtifile' parameter, events outside of these GTI intervals are excluded. Spectra are not accumulated across GTI intervals unless the 'spangti' parameter is set to 'yes'.

For each accumulated spectrum, 'ahgainfit' fits the theoretical profile to the data and also derives binned and unbinned averages. A least-squares method is used in the fitting. The fitted parameters are energy shift, scaling factor, background (unless the 'background' parameter is set to NONE), and, optionally, convolution width if 'fitwidth=yes'. The background is fit with a constant value if 'background' is set to CONSTANT, and a power-law if set to SLOPE. The unbinned average energy (as specified by 'energycol') is the sum of the energies in the spectrum divided by the number of events in the spectrum. The binned average energy is the weighted average derived by summing, over bins in the spectrum, the product of the energy and number of events per bin, and then dividing by the total number of events in the spectrum. The fitted gain corrections is computed from the fitted shift with respect to the theoretical line profile. The binned average gain correction is computed from the difference between the profile and spectrum averages.

The default values for the parameters used in the fitting method ('minwidth0', 'maxitcycle', 'r2tol', 'searchstepshift', 'maxdshift', 'bisectolshift', 'searchstepwidth', 'maxdwidth', 'bisectolwidth', and 'minwidth') should not need to change since already optimized.

The output file has two extensions. One extension, GRID_PROFILE, contains the energies and amplitudes of the theoretical profile used in the fitting procedure, including any convolution from the 'broadening' parameter. The other extension, DRIFT_ENERGY, reports the fitting results for each spectra in the following columns: TIME (midpoint of the time interval over which the spectrum is collected), splitcol (value of splitcol for spectrum as given by the 'splitcol' parameter; this column is absent if 'splitcol=NONE'), COR_FIT (energy correction factor from spectrum fit), COR_AVE (energy correction factor from spectrum average), CHISQ (reduced chi-squared of the fit), AVGUNBIN (average energy of events in spectrum prior to binning), AVGBIN (weighted spectrum average energy), AVGFIT (average energy from fit), SHIFT (fitted energy shift), SCALE (fitted vertical scaling factor), BGRND (fitted background), WIDTH (if 'fitwidth=no', same as broadening parameter; if 'fitwidth=yes', fitted width), TELAPSE (difference between times of first and last event in spectrum), EXPOSURE (calculated using the GTI), NEVENT (total number of events collected for this spectrum), BINMESH (array containing the count spectrum energy bins), SPECTRUM (array containing the observed binned count spectrum), FITPROF (array containing theoretical profile with fitted parameters applied). If the 'calcerr' parameter is set to 'yes', one-sigma errors for the SHIFT and WIDTH are calculated. The errors are calculated with chi-squared and maximum-likelihood methods and output in the columns SIGSHCHI2, SIGWDCHI2, SIGWDLIKE and SIGWDLIKE respectively. If 'writeerrfunc' parameter is set, the chi-squared and likelihood calculated valued are output in the arrays SHCHI2, SHLIKE, WDCHI2 and WDLIKE. The numbers of values output in these arrays are specified in the 'nerrshift' and 'nerrwidth' parameters, respectively.

PARAMETERS
infile [filename]
Input event file name. Input file may be a single file or a list of files (the name of the text file with the list preceded by the "@" character).

outevtsuffix [string]
Screened output file name suffix. This string is appended to the input file name (or name of first input file if a list).

outfile [filename]
Output gain correction file name.

(runselcal = no) [boolean]
If set to yes, run calibration to select calibration source events (yes/[no]).

(runsxipi = FULL) [string]
If runsxipi is set to FULL recalculate PI using the default mode of sxipi. If runsxipi is set to PARTIAL skip charge trail correction, CTI correction, and do not enable the split threshold iteration. If runsxsipi is set to no, do not run sxipi.

(rungtigen = no) [boolean]
If set to yes, run ahgtigen to create GTI (yes/[no]).

(runscreen = no) [boolean]
If set to yes, run ahscreen to filter input file (yes/[no]).

hkfile [filename]
Name of input SXI housekeeping (HK) file. If set to NONE, then this file is not used for the even-odd correction, and results in an error unless evnoddcor is set to no. This parameter is ignored if runsxipi is set to no.

(hkext = HK_SXI_USR_USER_HK1) [string]

HK extension name from which video temperature is read. This parameter is ignored if runsxipi is set to no.

(hkcolstem = SXI_USR_HKTBL) [string]
Column name stem where the video temperature is recorded in the HK file. This parameter is ignored if runsxipi is set to no.

(hkvideoid = A,B,B,B) [string]
Video card IDs used for the even-odd correction for CCD1, CCD2, CCD3, and CCD4. This parameter must consist of 4 characters of A or B separated with commas. This parameter is ignored if runsxipi is set to no.

(vtevnoddfile = CALDB) [filename]
Name of even-odd video temperature correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless evnoddcor is set to no. This parameter is ignored if runsxipi is set to no.

(chtrailfile = CALDB) [filename]
Name of charge trail correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless chtrailcor is set to no. This parameter is ignored if runsxipi is set to no.

(ctifile = CALDB) [filename]
Name of CTI correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless cticor is set to no. This parameter is ignored if runsxipi is set to no.

(spthfile = CALDB) [filename]
Name of split threshold values file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless spthcaldb is set to no. This parameter is ignored if runsxipi is set to no.

(gainfile = CALDB) [filename]
Name of gain file for PHA to PI conversion. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless gaincor is set to no. This parameter is ignored if runsxipi is set to no.

(patternfile = CALDB) [filename]
Name of file for grade hit pattern. If set to CALDB, the file is read from the calibration database. This parameter is ignored if runsxipi is set to no.

(randomize = yes) [boolean]
If set to yes, PHA is calculated from PHAS using decimal randomization. If startcol is PHAS_EVENODD, PHAS_TRAILCORR, or PHAS_CTICORR, this parameter is ignored, since those columns are floating point values. This parameter is ignored if runsxipi is set to no (yes/[no]).

(seed = -1457) [integer]
Random number generator seed; uses system time for seed=0. This parameter is ignored if runsxipi is set to no.

(gtigen_infile = mkf.fits) [filename]
Input mkf file used to create a GTI used in screening. This parameter is ignored if rungtigen is set to no.

(gtifile = NONE) [string]
Input gti file, or name of text file with list of files, preceded by "@" if including multiple input gti files to be merged. If the parameter is set to NONE no input GTI file is used. This parameter is ignored if rungtigen is set to no.

(selectfile = NONE) [string]
CALDB or user input label file with labels and expressions. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used.

(gtigenlabel = NONE) [string]
Labels to read from label file specified by selectfile and applied to gtigen_infile to create GTI, or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if rungtigen is set to no.

(screenlabel = NONE) [string]
Labels to read from label file specified by selectfile and used to screen input event file(s), or NONE. To input multiple labels, use a comma-separated list. This parameter is ignored if runscreen is set to no.

(gtiexpr = NONE) [string]
Expression, or label to read from label file specified by selectfile, used to create GTI. If the parameter is set to NONE no expression is used. This parameter is ignored if rungtigen is set to no.

(expr = NONE) [string]
Additional event screening expression used to screen input event file(s). If the parameter is set to NONE no expression is used. This parameter is ignored if runscreen is set to no.

(linefitfile = CALDB) [filename]
Input CALDB file containing parameters of the Lorentzian components used to construct the theoretical line profile. If the parameter is set to CALDB, the file is read from the calibration database.

(linetocorrect = Mnka) [string]
Calibration line to use for the gain correction. The value must match an extension in linefitfile. For the SXI this should be the SXI_MnK extension.

(energycol = UPI) [string]
Name of energy column to use in gain fitting.

(splitcol = PIXEL) [string]
Column name used to separate the data.

(numevent = 250) [integer]
Number of events collected for each spectrum used to calculate a single gain correction.

(minevent = 150) [integer]
Minimum number of events required for a spectrum. If the length of a group is less than minevent, those points are included with the previous group for processing, if possible.

(gapdt = -1) [real]
The upper limit to the time interval between two consecutive events in the same spectrum used in fitting. Two consecutive events separated in time by more than this amount are assigned to different groups. If gapdt=-1, no limit is imposed.

(grpoverlap = 0) [real]
The percentage overlap between adjacent groups. For grpoverlap=100 adjacent groups are shifted by one event, for grpoverlap=0 adjacent groups are independent and share no events.

(startenergy = -1.) [real]
Beginning of energy range in eV over which the spectra are collected. If startenergy is negative, the first energy is automatically determined by the smallest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(stopenergy = -1.) [real]
End of energy range in eV over which the spectra are collected. If stopenergy is negative, the final energy is automatically determined by the largest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(extraspread = 10.) [real]
Energy in eV by which the energy range is extended on either side beyond the smallest and largest energy in the linefitfile for the selected calibration feature. This parameter may be overridden by the startenergy and stopenergy parameters.

(evchannel = 1.) [real]
Conversion factor from channel number (for energycol) to energy [eV/chan].

(binwidth = 1.) [integer]
Energy bin width, in units of channels, to use when collecting spectra.

(broadening = 1.0) [real]
FWHM of the Gaussian in eV used to initially broaden the theoretical line profile. If fitwidth is set to no, the profile width is fixed at this value.

(gridprofile = no) [boolean]
If gridprofile is set to yes, only output the theoretical profile including any convolution due to the broadening parameter; no fitting is conducted (yes/[no]).

(fitwidth = no) [boolean]
If fitwidth is set to yes, then fit the width of each spectra in addition to the energy shift (yes/[no]).

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE).

(spangti = no) [boolean]
If spangti is set to yes, events in different intervals in gtifile may be collected in the same spectrum to be fit. If spangti is set to no, then groups of events used to construct the spectra must be from the same GTI. This parameter is ignored if gtifile is set to NONE (yes/[no]).

(avgwinrad = -1.) [real]
Radius of interval (in units of binwidth) used only to update the initial shift estimate prior to fitting. If avgwinrad is set to -1, this radius is automatically calculated based on the theoretical line profile and the broadening parameter. This is not used in calculating the average results.

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no]).

(writeerrfunc = no) [boolean]
Output the array of chi-squared and likelihood calculated for the SHIFT and WIDTH (yes/[no]).

(minwidth0 = 1.0) [real]
Smallest width, in units of binwidth, allowed as the initial value in width fitting. This parameter provides a lower limit to the initial estimate of the width as computed by the fitting algorithm. The value must be greater than zero.

(maxitcylce = 5) [integer]
Maximum number of fitting iterations.

(r2tol = .01) [real]
Convergence criterion on R-squared for least-squared fitting. Once R-squared changes by less than this amount between fitting iterations, the procedure is finished. This parameter should not normally need to be changed from the default value.

(searchstepshift = 2.) [real]
Step size, in units of binwidth, used when searching for best-fit energy shift in either direction from the initial shift estimate based on the spectrum average. The final shift is obtained using the bisection method (see bisectolshift).

(maxdshift = 5.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of energy shift. If no solutions are found within this deviation at smaller or larger shifts from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolshift = .1) [real]
When the bisection method determines the energy shift to within this amount in units of binwidth, the fitting procedure is completed.

(searchstepwidth = 5.) [real]
Step size, in units of binwidth, used when searching for best-fit convolution width in either direction from the initial width estimate based on the difference between the profile and spectrum statistical variances. The final width is obtained using the bisection method (see bisectolwidth).

(maxdwidth = 10.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of convolution width. If no solutions are found within this deviation at smaller or larger widths from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolwidth = .2) [real]
When the bisection method determines the convolution width to within this amount in units of binwidth, the fitting is procedure is completed.

(minwidth = .5) [real]
Since the least-squares fitting functional is undefined when the width is zero, one must define a minimum allowed fitted width. If the fitting routine attempts to fit a width smaller than this value (in units of binwidth), the fitting procedure fails for the spectrum.

(nerrshift = 100) [integer]

Number of shift values in uncertainty calculations.

(nerrwidth = 100) [integer]
Number of width values in uncertainty calculations.

(shifterrfac = 3.0) [real]
Factor for determining domain of shift uncertainty arrays.

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Compute the gain correction for an SXI event file using Mn K lines as the theoretical profile. The theoretical profile is convolved with a 12 eV Gaussian and gains are computed separately for each CCD_ID.
sxigainfit infile=event_in.fits outfile=drift_out.fits linetocorrect=SXI_Mnk energycol=PI splitcol=CCD_ID extraspread=20. \
evchannel=6. broadening=12. fitwidth=yes

NAME sximodegti -- Create GTI excluding dead time for each SXI observing mode

USAGE sximodegti infile outfile mergefile tstart tstop

DESCRIPTION
'sximodegti' creates a Good Time Interval (GTI) for each supported SXI mode (full window, 1/8 window, full window+burst, 1/8 window+burst, 1/8 window+area discrimination). These GTIs are used for exposure map generation. The task reads as input the exposure FFF file, and creates GTI(s) for all the modes found.
The SXI exposure file for an observation contains rows of exposure packets that corresponds to a 3x3 and 5x5 event packet (contained separately in the event list). Each CCD and segment combination is contained in one row in the exposure, so there should be 8 rows per time stamp in this file. However, the file contains information for all 'modes' run during the observation, including different DATACLASS (operating configuration), events, iframe, dframe, and rframe. Only some of these are supported for science data. This script reads the exposure file and creates a GTI extension for each combination of CCD_ID, SEGMENT, and supported science mode, keeping track of the total number of frames for proper calculation of the exposure time.

There are two output files whose names are set by the parameters 'outfile' and 'mergefile'. The first output 'outfile' contains a GTI extension for each DATACLASS and frame. Each extension contains START, STOP, and frame number (FRAMENUM) columns. The task also checks if a CCD_ID+SEGMENT combination is missing or appears more than once. The information is output to the logfile of the task. The second output 'mergefile' contains a GTI extension per DATACLASS, where GTIs for different CCD_ID+SEGMENT combinations are merged into one (using a logical 'AND'). The number of unique DATACLASS values is also output to the parameter 'numdclass'.

Only times within the range specified by the 'tstart' and 'tstop' parameters are included in the output GTIs. Normally these are the start and stop times of pointed observations, to exclude the slew data. If the parameters 'tstart' or 'tstop' are outside of the range covered by the exposure file, then the file TSTART or TSTOP header keywords are used instead.

PARAMETERS
infile [filename]
Input SXI exposure file.

outfile [filename]
Output GTI file. This file has one extension per combination of DATACLASS, CCD_ID, and SEGMENT.

mergefile [file]
Output GTI file with merged dataclasses. This file has one extension per DATACLASS.

tstart [real]
Start time for pointed observations [sec]. If the TSTART of 'infile' is later than this value, then that TSTART is used instead.

tstop [real]
Stop time for pointed observations [sec]. If the TSTOP of 'infile' is earlier than this value, then that TSTOP is used instead.

(numdclass = 0) [integer]
Output: Total number of dataclasses found.

(cleanup = yes) [boolean]
If set to yes, temporary files are deleted ([yes]/no).

[ccldhm]

EXAMPLES
1. Run sximodegti with default parameters.
sximodegti infile=ah00050733sxi_a0exp.fits.gz outfile=ah000507033sxi_mode.gti.gz \
mergefile=ah000507033sxi_seg.gti.gz tstart=47390384. tstop=47430516.

NAME sxinxbgen -- Create a Non-X-ray Background (NXB) spectrum for SXI

USAGE sxinxbgen infile ehkfile regfile innxbfile innxbehk innxbhk outpifile

DESCRIPTION
'sxinxbgen' generates a PI spectrum of the non-X-ray background (NXB) for an SXI observation. The NXB spectrum is calculated from a calibration file containing night Earth data, when X-rays are blocked from the detector's view, accumulated over a period of time. This spectrum is used in spectral fitting (e.g. with XSPEC) to subtract the effects of the particle background from the science data.

'sxinxbgen' uses the following inputs: (a) the science SXI event file for which the NXB spectrum is calculated ('sxinxbgen' reads and uses information in the header keywords related to the time of the observation, and the pointing, as well as the GTI); (b) the extended housekeeping (EHK) file ('ehkfile') containing information on the spacecraft orbit during the observation; (c) The NXB event file (parameter 'innxbfile'); the NXB EHK file ('innxbehk'); (d) a DS9-format region file ('regfile') describing the spectral extraction SXI source region. The region may be specified in RAW, DET, FOC, or SKY coordinates (see 'regmode' parameter). 'sxinxbgen' includes options for recalculating the PI values in the input NXB event file (see 'apply_sxipi' parameter), and for excluding times of high background by extracting a light curve from the NXB data (see 'runlcurve' parameter).

If 'apply_sxipi=yes', 'sxinxbgen' first runs the tool 'sxipi' to apply the latest calibration to the NXB events before filtering and extracting the spectrum. 'sxipi' requires an SXI housekeeping (hk) file for the observation (see 'innxbhk' parameter). The 'sxipi' parameters 'hkfile', 'hkext', 'hkcolstem', 'hkvideoid', 'vtevnoddfile', 'chtrailfile', 'ctifile ', 'spthfile', 'gainfile', 'patternfile', 'startcol', 'evnoddcor', 'badpixopt', 'evtthre', 'deltatime', 'randomize', and 'seed' may be set within 'sxigainfit' (see 'sxipi' for details).

'sxinxbgen' next filters the NXB event data on the region file. This selection is done using DET coordinates and the task internally transforms the coordinates if the region file is not provided in DET. The task then creates a GTI to apply to the NXB event and NXB EHK data using the 'timefirst' and 'timelast' parameters. These extend the NXB GTI beyond that of the science data to ensure sufficient statistics in the output NXB spectrum. A baseline default value, based on experience with previous missions, is to have a window of 300 days centered on the observation. 'sxinxbgen' then screens the NXB events using the same criteria used for selection of events in the science data (see 'expr' parameter). If 'runlcurve=yes', a light curve is extracted in the 12-24 keV range from the NXB data and used at this point to remove times of high background. Finally the NXB spectrum is produced from NXB events within this GTI that are selected and weighted based on the geomagnetic Cut-Off Rigidity (COR), an estimate of the shielding provided by the Earth's magnetic field against impinging charged particles. NXB spectra are extracted based on the distribution of COR that are present in the science event data and weighted by the ratio of the science exposure time in that bin to the total science exposure time. The NXB output spectrum is then the sum of these weighted spectra.

The EHK file contains several Cut-Off Rigidity values. Empirically derived values are stored in the columns COR, COR2, or COR3; and a calculated value in the CORTIME column. The choice of COR to use is specified by the 'sortcol' parameter. COR3 is the recommended table to use for SXI. The 'sortbin' parameter specifies the COR value bin boundaries that are used in the NXB selection.

'sxinxbgen' outputs the weighted NXB PI spectrum ('outpifile'); and optionally, the EHK file corresponding to the science GTI ('outehkfile'); the calibrated, screened, and time-filtered NXB event list ('outnxbfile'); and the NXB EHK file corresponding to that NXB file ('outnxbehk').

PARAMETERS
infile [filename]

Input event file. Header keywords are read from the EVENTS extension, and the GTI extension is used to construct the weighting histogram.

ehkfile [filename]
Input EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

regfile [filename]
Input region file in DS9 format. The region should be in coordinates specified by regmode.

innxbfile [filename]
Input NXB event file used to construct the NXB spectrum.

innxbehk [filename]
Input NXB EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

innxbhk [filename]
Input NXB HK file. Contains HK data needed to reprocess the NXB data with sxipi, as specified by the hkext, hkcolstem, and hkvideoid parameters.

outpifile [filename]
Output PI spectrum file name.

(outehkfile = NONE) [filename]
Output EHK file. Contains the EHK data for only the times in the infile GTI. If the parameter is set to NONE this file is not created.

(outnxbfile = NONE) [filename]
Output NXB file. Contains the filtered NXB events used in the output spectrum. If the parameter is set to NONE this file is not created.

(outnxbehk = NONE) [filename]
Output NXB EHK file. Contains the EHK data for only the times covered by the filtered NXB events. If the parameter is set to NONE this file is not created.

(regmode = SKY) [string]
Region mode. Specifies the coordinate system used by regfile. Allowed systems are SKY, DET, FOC, and RAW.

(timefirst = 150) [integer]
Days before the science observation used to extract NXB.

(timelast = 150) [integer]
Days after the science observation used to extract NXB.

(sortcol = COR3) [string]
Column for sorting NXB data. This column must exist in the EHK files, and must be either COR, COR2, COR3, or CORTIME.

(sortbin = 0,4,5,6,7,8,9,10,11,12,13,99) [string]
Bin boundaries for sorting NXB data. Times where sortcol is below the minimum or above the maximum value are excluded. List must be comma-separated in increasing numerical order. The range should match any COR filtering performed on the science data.

(expr = NONE) [string]
Additional expression to select good events applied to the NXB event list. This should match the screening of the science data. If the parameter is set to NONE no expression is used.

(runlcurve = no) [boolean]
If runlcurve is set to yes a light curve is extracted in the 12-24 keV range from the NXB data and used to exclude times of high background; if set to no lightcurve filtering is not conducted (yes/[no]).

(apply_sxipi = yes) [boolean]
If apply_sxipi is set to yes recalculate PI using sxipi. If apply_sxsipi is set to no, do not run sxipi ([yes]/no).

(hkext = HK_SXI_USR_USER_HK1) [string]
HK extension with video temperatures. This parameter is ignored if apply_sxipi is set to no.

(hkcolstem = SXI_USR_HKTBL_) [string]
Column template for video temperatures. This parameter is ignored if apply_sxipi is set to no.

(hkvideoid = A,B,B,B) [string]
Video card ID for gain correction of CCD1-4. This parameter is ignored if apply_sxipi is set to no.

(vtevnoddfile = CALDB) [filename]
Input video evenodd file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used. This parameter is ignored if apply_sxipi is set to no.

(chtrailfile = CALDB) [filename]
Input charge trail correction file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used. This parameter is ignored if apply_sxipi is set to no.

(ctifile = CALDB) [filename]
Input CTI correction file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used. This parameter is ignored if apply_sxipi is set to no.

(spthfile = CALDB) [filename]
Input split threshold file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used. This parameter is ignored if apply_sxipi is set to no.

(gainfile = CALDB) [filename]
Input gain correction file. If the parameter is set to CALDB, the file is read from the calibration database. This parameter is ignored if apply_sxipi is set to no.

(patternfile = CALDB) [filename]
Input grade hit pattern file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE this file is not used. This parameter is ignored if apply_sxipi is set to no.

(startcol = PHAS) [string]
The column name with which to start the corrections. The allowed values are PHAS, PHAS_EVENODD, PHAS_TRAILCORR, PHAS_CTICORR, or PHA. All but PHAS and PHA are only present if the event file has already been processed with sxipi with debugcol=yes. This parameter is ignored if apply_sxipi is set to no.

(evnoddcor = yes) [boolean]
If set to yes, the even-odd correction is enabled. This parameter is ignored if apply_sxipi is set to no ([yes]/no).

(badpixopt = 2) [integer]
Option specifying how to process events that contain bad pixels. Must be an integer in the range 1-3. If set to 1, bad pixels are ignored in the 3x3 and all PHAS values are used to grade and calculate the summed PHA. This is the method used by the Suzaku XIS. If set to 2, the task applies PHAS_MASK to mask out bad pixels in the 3x3 and set them to NULL. Grading and PHA summation are then done normally, effectively ignoring these pixels. If set to 3, the task applies PHAS_MASK to mask out bad pixels in the 3x3 and set them to NULL. The GRADE, PHA, and PI values are also set to NULL and these are considered bad events. This is the most conservative option, ensuring better fidelity at the expense of fewer counts. This parameter is ignored if apply_sxipi is set to no.

(evtthre = DEFAULT) [string]
Event threshold value. If set to DEFAULT, the task uses the values in the infile header keyword EVENTTHR, which contains one value for each CCD_ID and SEGMENT combination. Otherwise, a single numerical floating-point value can be specified, and is used for all CCD_IDs and SEGMENTs. This parameter is ignored if apply_sxipi is set to no.

(deltatime = 8) [integer]
Acceptable time gap to search for video temperature in HK file. If a time entry is not found within this deltatime of the event, then an error flag is assigned in the STATUS. This parameter is ignored if apply_sxipi is set to no.

(randomize = yes) [boolean]
If set to yes, PHA is calculated from PHAS using decimal randomization. If startcol is PHAS_EVENODD, PHAS_TRAILCORR, or PHAS_CTICORR, this parameter is ignored, since those columns are floating point values. This parameter is ignored if apply_sxipi is set to no (yes/[no]).

(seed = -1457) [integer]
Random number generator seed; uses system time for seed=0. This parameter is ignored if apply_sxipi is set to no.

(cleanup = yes) [boolean]
Delete intermediate files ([yes]/no).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Run sxinxbgen with a restrictive COR range, to reduce the background for a low-surface-brightness target.
sxinxbgen infile=target.evt ehkfile=ehk.fits regfile=target_sky.reg innxbfile=nxb.fits \
innxnehk=nxb.ehk innxbhk=nxb.hk outpifile=target_sxinxb.pi sortbin="6,7,8,9,10,11,12,13,99"

2. Run sxinxbgen using a region in DET coordinates, and excluding the calibration source regions.
sxinxbgen infile=target.evt ehkfile=ehk.fits regfile=target_det.reg innxbfile=nxb.fits\
innxbehk=nxb.ehk innxbhk=nxb.hk outpifile=target_sxinxb.pi regmode=DET expr="STATUS[2]==b0"

3. Run sxinxbgen without applying the most recent SXI calibration.
sxinxbgen infile=target.evt ehkfile=ehk.fits regfile=target_sky.reg innxbfile=nxb.fits\
innxbehk=nxb.ehk innxbhk=nxb.hk outpifile-target_sxinxb.pi apply_sxipi=no

---
NAME sxiphas -- Merge inner 3x3 and outer 5x5 pulse height columns in SXI event list

USAGE sxiphas infile outfile

DESCRIPTION
'sxiphas' reads an SXI FITS event list and merges pixel pulse height data from the inner 3x3 event island (9 pixels in the column PHAS_INNER3X3) and the outer 5x5 ring (16 pixels in the column PHAS_OUTER5X5) for each event. Since the inner 3x3 square are telemetered separately from the outer 5x5 ring, they must be combined into a single column. Also, depending on the telemetry bandwidth available, one or the other set of pulse heights might be missing.

The merged pulse height data from all 25 pixels are recorded in the column PHASALL. The contents of PHAS_INNER3X3 are also copied to the PHAS column. If the 3x3 data are present but the 5x5 data are missing, then the outer pixel values of PHASALL are set to TNULL, and STATUS flag 24 is set. If the inner 3x3 data are missing (which is abnormal), then regardless of whether the outer 5x5 ring is present, all of PHAS and PHASALL are set to TNULL, and STATUS flag 25 is set.

PARAMETERS
infile [filename]
Name of input SXI FITS event list.

outfile [filename]
Name of output SXI FITS event list.

(colbound = -32768,-32767,32767) [string]
Values to use for TNULL, TLMIN, and TLMAX in the output columns PHASALL and PHAS. TNULL is the value used for "no data". This is a comma-separated list of three integers.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Merge 3x3 and 5x5 pulse height columns, using the default TNULL, TLMIN, and TLMAX values for the output pulse height amplitude columns PHAS and PHASALL.
sxiphas infile=sxi_fff_test_20130418.fits outfile=sxi_sff_test_20130418.fits

---
NAME
sxipi -- Calculate pulse invariant (PI) values and assign grades for SXI events.

USAGE
sxipi infile outfile hkfile
DESCRIPTION

'sxipi' calculates a pulse invariant (PI), a value proportional to the X-ray energy, for each event in an SXI event list by applying several pulse height corrections and a grade assignment. There are 5 major steps in this task:

1. Video board temperature (or even-odd column) correction
2. Charge trail correction
3. Charge transfer inefficiency (CTI) correction
4. Grade assignment and PHA summation;
5. PHA to PI conversion (gain correction)

In Step 1, differences in PHAS between even and odd CCD columns are corrected. Each CCD chip is associated with two circuit boards (ASICs) where the raw pixel output is amplified and digitized. To speed up the signal processing, one of the ASICs reads signals transferred from the even-RAWX pixels, while the other reads those from the odd-RAWX pixels. The conversion of analog to digital signal can differ for different ASICs, and this conversion depends on the temperature of the video board at the location of the ASIC. This step removes those differences, correcting the pixel pulse height values so they are on the same scale for a given event. The video board temperature is read from the HK file specified by the parameter 'hkfile', with other parameters 'hkext', 'hkcolstem', and 'hkvideoid' specifying where in the file that information is. Step 1 can be turned on and off with the 'evnoddcor' parameter.

In Steps 2 and 3, the task corrects the pulse height information of individual pixels to account for charge lost during the transfer. When the detected events are transferred for readout, some amount of charge is trapped by defects in the CCD chips, which are mainly caused by radiation damage. Some of the trapped charge is released immediately after a one-pixel transfer, resulting in a 'trail' into the following pixel. Long-timescale charge traps release the charge in pixels outside of the event island, effectively losing this information. Charge injection was developed as a way to mitigate CTI by periodically filling traps with rows of 'sacrificial' charge. As a result, the CTI strongly depends on the configuration of charge injection, and the task reads that configuration information from the event file header keywords. Step 2 can be turned on and off with the 'chtrailcor' parameter, and Step 3 with the 'cticor' parameter.

The CTI calibration for SXI depends on the event GRADE, however this information is not calculated until Step 4. If the parameter 'ctigrade=yes', the GRADE column of the input event list is used in the CTI correction. If 'ctigrade=no', the GRADE column is ignored. Proper CTI correction requires an initial run of sxipi with 'ctigrade=no' to populate the GRADE column, and then a second run of sxipi with 'ctigrade=yes' on the output file of the first run. The parameter 'copygrade' can be used to copy the original GRADE and PHA columns to GRADE_COPY and PHA_COPY columns before they are overwritten, to compare results with and without grade-dependent CTI correction.

In Step 4, a GRADE is assigned to each event based on the number and position of surrounding pixels whose pulse height exceeds the split threshold. These pulse heights are recorded for individual 3x3 pixels in the PHAS column, and the hit pattern of the outer 5x5 (16 pixel) ring is also used by way of the P_OUTER_MOST column. The total event pulse height amplitude (PHA) is calculated by summing up the PHAS in the pattern. Step 4 is always performed and cannot be turned off.

In the default usage, the split threshold is energy-dependent, optimized depending on the tentative PHA value. A split threshold is optimized for each event in the following way: (1) a tentative GRADE is assigned using an initial fixed split threshold; (2) a tentative summed PHA is calculated; (3) the optimum split threshold is determined; and (4) the final GRADE assignment and PHA value are determined. There are 4 parameters related to this optimization process, 'spthiter', 'spthcaldb', 'spthoffset', and 'spthslope'. The default values of the first two parameters are both 'yes'. In this case, the values in the last two parameters are ignored, and the split threshold optimization proceeds normally. In the case of 'spthcaldb=no', the values of 'spthoffset' and 'spthslope' are read and used, instead of the CALDB values, for the optimization. If 'spthiter=no', the iteration described above is not performed, and a constant split threshold from CALDB (if 'spthcaldb=yes') or 'spthoffset' (if 'spthcaldb = no') is used for the grade assignment. The 'spthslope' parameter is ignored in both cases with 'spthiter=no'.

In Step 5, PHA is converted to PI using the gain correction table. PI is directly proportional to the X-ray energy, with 1 PI channel = 6 eV for SXI. Step 5 can be turned on and off with the 'gaincor' parameter. If turned off, PHA is simply copied to PI.

Bad pixels in the surrounding 3x3 ring can alter the grade and final PI of an event, possibly causing bad (e.g. particle) events to migrate to good events, thus losing information about the true event pulse height. The 'badpixopt' parameter, along with the PHAS_MASK column (see the sxiflagpix help) can be used to mitigate this by specifying how events with bad pixels should be graded and summed. There are three options, which are listed in the parameter description below.

For debugging and calibration purposes, the corrections in Steps 1-3 and 5 can be turned on or off independently using the 'evnoddcor', 'chtrailcor', 'cticor', and 'gaincor' parameters. Event grades and summed PHA values are always calculated. Furthermore, the intermediate results from each correction can be included in the output by specifying 'debugcol=yes'. Any of these debugging columns or the PHAS

or PHA column can then be used as the starting point for another run of sxipi with the 'startcol' parameter. It is important to note that any corrections turned on with 'evnoddcor', 'chtrailcor', 'cticor', and 'gaincor' are performed regardless of the 'startcol', and so it is possible to perform the same correction twice. If 'startcol=PHA', then all corrections that are enabled are applied to the PHAS column, but the gain correction in Step 5 is applied to the PHA column in the input file. For processing of normal science data, these features should not be used, and these parameters should be kept at their default values of 'yes'.

This task sets the following STATUS flags if an event satisfies certain conditions, shown in the table below. See the 'sxiflagpix' help file for a full list of STATUS flags.

flag	description
26	(sxipi - general) PHAS[0] < event threshold
27	(sxipi - vtevnodd) Video temperature is out of range
28	(sxipi - vtevnodd) Lack of video temp HK at time close to the event
29	(sxipi - chtrail/CTI) Correction value is negative
30	(sxipi - general) Null value by correction process

PARAMETERS

infile [filename]
Name of input SXI FITS event list.

outfile [filename]
Name of output SXI FITS event list.

hkfile [filename]
Name of input SXI housekeeping (HK) file. If set to NONE, then this file is not used for the even-odd correction, and results in an error unless 'evnoddcor=no'.

(hkext = HK_SXI_USR_USER_HK1) [string]
HK extension name from which video temperature is read.

(hkcolstem = SXI_USR_HKTBL) [string]
Column name stem where the video temperature is recorded in the HK file.

(hkvideoid = A,B,B,B) [string]
Video card IDs used for the even-odd correction for CCD1, CCD2, CCD3, and CCD4. This parameter must consist of 4 characters of A or B separated with commas.

(vtevnoddfile = CALDB) [filename]
Name of even-odd video temperature correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless 'evnoddcor=no'.

(chtrailfile = CALDB) [filename]
Name of charge trail correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless 'chtrailcor=no'.

(ctifile = CALDB) [filename]
Name of CTI correction file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless 'cticor=no'.

(spthfile = CALDB) [filename]
Name of split threshold values file. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless 'spthcaldb=no'.

(gainfile = CALDB) [filename]
Name of gain file for PHA to PI conversion. If set to CALDB, the file is read from the calibration database. If set to NONE, the task does not use this calibration information, and results in an error unless 'gaincor=no'.

(patternfile = CALDB) [filename]

Name of file for grade hit pattern. If set to CALDB, the file is read from the calibration database.

(startcol = PHAS) [string]
The column name with which to start the corrections. The allowed values are PHAS, PHAS_EVENODD, PHAS_TRAILCORR, PHAS_CTICORR, or PHA. All but PHAS and PHA are only present if the event file has already been processed with sxipi with 'debugcol=yes'.

(evnoddcor = yes) [boolean]
If set to yes, the even-odd correction is enabled ([yes]/no).

(chtrailcor = yes) [boolean]
If set to yes, the charge trail correction is enabled ([yes]/no).

(cticor = yes) [boolean]
If set to yes, the CTI correction is enabled ([yes]/no).

(gaincor = yes) [boolean]
If set to yes, the gain correction is enabled ([yes]/no).

(ctigrade = yes) [boolean]
If set to yes, use of grade information in the CTI correction is enabled ([yes]/no).

(copygrade = no) [boolean]
If set to yes, the existing GRADE and PHA columns to are copied to GRADE_COPY and PHA_COPY (respectively) before overwriting them with newly-calculated values (yes/[no]).

(phcut = CALDB) [string]
Pulse-height cut for the CTI correction (in ADU). If set to CALDB, value from the header of the CTI CALDB file 'ctifile' is used.

(badpixopt = 2) [integer]
Option specifying how to process events that contain bad pixels. Must be an integer in the range 1-3. If set to 1, bad pixels are not flagged in the 3x3 and all PHAS values are used to grade and calculate the summed PHA. This is the method used by the Suzaku XIS. If set to 2, the task applies PHAS_MASK to mask out bad pixels in the 3x3 and set them to NULL. Grading and PHA summation are then done normally, effectively ignoring these pixels. If set to 3, the task applies PHAS_MASK to mask out bad pixels in the 3x3 and set them to NULL. The GRADE, PHA, and PI values are also set to NULL and these are considered bad events. This is the most conservative option, ensuring better fidelity at the expense of fewer counts.

(spthiter = yes) [boolean]
If set to yes, the grade assignment is iterated to optimize the split threshold ([yes]/no).

(spthcaldb = yes) [boolean]
If set to yes, the split threshold values found in the 'spthfile' file are used ([yes]/no).

(spthoffset = 15.) [float]
Split threshold offset value, used when 'spthcaldb=no'.

(spthslope = 0.) [float]
Split threshold slope value, used when 'spthcaldb=no'.

(evtthre = DEFAULT) [string]
Event threshold value. If set to DEFAULT, the task uses the values in the infile header keyword 'EVENTTHR', which contains a comma-separated list of values for each CCD_ID and SEGMENT combination. Otherwise, a single numerical floating-point value can be specified, and is used for all CCD_IDs and SEGMENTs.

(negthre = -5000) [integer]
Minimum value of any PHAS array element allowed for a normal event.

(deltatime = 8) [integer]
Acceptable time gap to search for video temperature in HK file. If a time entry is not found within this 'deltatime' of the event, then an error flag is assigned in the STATUS.

(debugcol = no) [boolean]

If set to yes, the intermediate values after each correction step is output (yes/[no]). This option can be used for debugging and calibration. The columns written are PHAS_EVENODD, PHAS_TRAILCORR, PHAS_CTICORR, PHA_SPTH, and GRADE_SPTH.

(randomize = yes) [boolean]
If set to yes, randomization is used to convert the input integer PHAS or PHA column to floating point ([yes]/no). If 'startcol' is 'PHAS_EVENODD', 'PHAS_TRAILCORR', or 'PHAS_CTICORR', this parameter is ignored, since those columns are floating point values.

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Run sxipi with default parameters
sxipi infile=ah_sxi_input.evt outfile=ah_sxi_output.evt hkfile=ah123456789sxi.hk

2. Run sxipi disabling the CTI correction
sxipi infile=ah_sxi_input.evt outfile=ah_sxi_output.evt hkfile=ah123456789sxi.hk cticor=no

3. Run sxipi using a constant split threshold of 20.
sxipi infile=ah_sxi_input.evt outfile=ah_sxi_output.evt hkfile=ah123456789sxi.hk spthiter=no spthcaldb=no spthoffset=20

4. Run sxipi with debugging columns output.
sxipi infile=ah_sxi_input.evt outfile=ah_sxi_output.evt hkfile=ah123456789sxi.hk debugcol=yes

5. Run sxipi on the previous output, starting with the charge trail correction and only applying that. This requires that the debugging columns are present in the input file, so sxipi must have been run previously.
sxipi infile=ah_sxi_sxipi1_debug.evt outfile=ah_sxi_sxipi2_debug_onlychtrail.evt hkfile=ah123456789sxi.hk \
startcol=PHAS_EVENODD evnoddcor=no chtrailcor=yes cticor=no debugcol=yes

6. Run sxipi, skipping the even odd, charge trail, and CTI corrections. This would be useful for verifying the gain (PHA to PI conversion) calibration quality alone. Debug columns are also output to verify the proper corrections were made (or not).
sxipi infile=ah_sxi_input.evt outfile=ah_sxi_output_onlygain.evt hkfile=ah123456789sxi.hk evnoddcor=no chtrailcor=no cticor=no \
debugcol=yes

NAME sxipipeline - SXI reprocessing tool

USAGE sxipipeline indir outdir steminputs stemoutputs entry_stage attitude orbit housekeeping

DESCRIPTION
sxipipeline duplicates most of the pipeline (not trend data) for the SXI. It allows the user to run all or part of the pipeline processing and to vary the calibration files and filtering (screening) criteria used. A number of other pipeline processing parameters can also be changed.

SXI Pipeline Stages

The sxipipeline is divided into 3 stages:

Calibration
Data screening
Product creation

Each stage may be run singly or in combination with the preceding stages. This is controlled by the entry_stage and exit_stage parameters.

SXI Stage 1 consists of the following steps:
For each event file:
(optional) Run coordevt on hot pixel file

Run coordevt
Run sxiphas
Run sxiflagpix
Run sxipi
Run sxipi
Create flickering pixel file
Filter SXI events STATUS[1]==b0
Run searchflickpix
Run coordevt on flickering pixel events
Run sxiflagpix with flickering pixels

The data screening (Stage 2) is identical to that in the production pipeline, when default parameters are used. For details on the default screening applied to the SXI events (respectively), see:
ahfilter - Create the EHK from attitude & orbital data and create the MKF from housekeeping data based on the CALDB mkfconf file
ahgtigen - Create the GTI from the EHK and MKF parameters based on CALDB selection file
ahscreen - Screen the data based on GTI and CALDB selection file
Default GTI used for screening data are:
GTIPOINT
GTITEL
GTI_<DATACLASS>
GTIEHK
GTIMKF
Optionally, sximodegti can be run to create new GTI specific to the DATACLASSes of the input SXI event file.

The product creation (Stage 3) is identical to that in the production pipeline, when default parameters are used. For SXI events extractor is run on SKY coordinates and a lightcurve, spectra and images are created for each cleaned event file. No gif images are created.

INPUT
The input to sxipipeline is specified using (at minimum) the indir parameter. This should be specified as the sxi event_uf level sequence directory, e.g.:

sxipipeline indir=/path/to/100039010/sxi/event_uf

Paths to specific sxi housekeeping and satellite data files can be specified using the attitude, housekeeping, extended_housekeeping, makefilter, orbit and obsgti parameters. The attitude, orbit and sxi housekeeping files are required for stage 1 calibration.

OUTPUT
Filenames, etc.

The number of output files depends on both pipeline processing stage(s) and the options selected. All output files are written to the directory specified by the outdir parameter. The archive directory structure is NOT reproduced (i.e. all output files are in a single directory).

The names of files produced are the same as those found in the HEASARC archive. However the usual "ahXXXXXXXXX" prefix, where "XXXXXXXXX" is the sequence number, can be replaced by a character string set by the stemoutputs parameter. This defaults to the value set by the steminputs parameter.

PARAMETERS
indir [string]
Directory containing the input data. This should be the path to the event_uf (or event_cl for stage_3 processing)
;/path/to/805062010/sxi/event_uf

outdir [string]
Output directory used for all output data, as well as the report file.
CAUTION: If clobber is set, and this directory already exists, then this task overwrites files as needed in this directory.

steminputs [string]
Stem for FITS input files, e.g. ah100039010. This string is used as the base filename for finding input files. For example, if steminputs=ah100039010, then to find the attitude file, ahpipeline matches the regular expression /ah100039010sxi_[ps][\d][0-9a-f]{8}_uf\.evt(\..+)?$/ against all files found in the indir directory.

(stemoutputs = DEFAULT) [string]

Base (stem) output name used for creating output files. If set to "DEFAULT", then steminputs is used.

entry_stage = 1 [1|2|3]
Entry stage, 1 or 2.
Stage 1: Re-calibrate unfiltered event files.
Stage 2: Start from existing unfiltered event files.

exit_stage = 2 [1|2|3]
Exit stage, 1 or 2.
Stage 1: Produces calibrated unfiltered event files.
Stage 2: Produces screened event files.

attitude [string]
Attitude file

(extended_housekeeping = ah1001.ehk) [string]
Extended housekeeping file

(makefilter = ah1001.mkf) [string]
Makefilter file

orbit [string]
Orbit file

gtifile [string]
Input GTI file

housekeeping [string]
SXI Housekeeping file

(regionfile = NONE) [string]
Input region file

(sxi_mkflabel = SXI#SCI) [string]
Label to use for SXI MKF GTI creation. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxi_ehklabel = SXI#SCI) [string]
Label to use for SXI EHK GTI creation. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxi_evtlabel = SXI#SCI) [string]
Label to use for SXI event screening. The hash is replaced by a respective mode: CCDW1, CCD12WA or CCD34WF

(sxi_start = 0.0) [real]
SXI CALDB start time

(ra = -999.99999) [real]
RA of nominal pointing [deg]

(dec = -999.99999) [real]
Dec of nominal pointing [deg]

(roll = 0.0) [real]
Roll of nominal pointing [deg]

(optdetx = -999.99999) [real]
SXI optical detx coordinate

(optdety = -999.99999) [real]
SXI optical dety coordinate

(optfocx = -999.99999) [real]
SXI optical focx coordinate

(optfocy = -999.99999) [real]
SXI optical focy coordinate

(optskyx = -999.99999) [real]
SXI optical skyx coordinate

(optskyy = -999.99999) [real]
SXI optical skyy coordinate

(ra_pnt = -999.99999) [real]
RA of sxi pointing [deg]

(dec_pnt = -999.99999) [real]
DEC of sxi pointing [deg]

(calc_hotpix = no) [boolean]
Run coordevt on hot pixel file (yes/[no])

(calc_modegti = yes) [boolean]
Calculate SXI data mode GTI ([yes]/no)

(teldeffile = CALDB) [string]
Input teldef file (or CALDB)

(leapsecfile = REFDATA) [file]
ahscreen: Input leap second file (or CALDB, [REFDATA])

(selectfile = CALDB) [file]
ahscreen: Input file with the selection expressions

(badpixfile = CALDB) [file]
sxiflagpix: badpixfile

(maskfile = CALDB) [file]
sxiflagpix: maskfile

(vtevnoddfile = CALDB) [file]
sxipi: evenodd

(ctifile = CALDB) [file]
sxipi: cti

(chtrailfile = CALDB) [file]
sxipi: chtrail

(spthfile = CALDB) [file]
sxipi: splitth

(gainfile = CALDB) [file]
sxipi: gain

(patternfile = CALDB) [file]
sxipi: grade

(dattfile = datt.out) [string]
output datt file with drift corrections

(coordevt_startsys = LOWEST) [string]
Starting coordinate system

(stopsys = HIGHEST) [string]

Final coordinate system

(annaber = no) [string]
Apply annual aberration correction (yes, [no], INVERT)

(followsun = no) [boolean]
Recalculate the Sun position for each event (yes, [no])

(orbaber = no) [string]
Apply sat orbital aberration correction (yes, [no], INVERT)

(attinterp = LINEAR) [string]
Sky attitude interpolation method (LINEAR, CONSTANT)

(dattinterp = LINEAR) [string]
Delta attitude interpolation method (LINEAR, CONSTANT)

(attdt = 32.) [real]
Allowed margin for time extrapolation in attfile [s]

(dattdt = 0.5) [real]
Allowed margin for time extrapolation in dattfile [s]

(chkattgap = no) [boolean]
Limit attitude interpolation if gaps present (yes, [no])

(chkdattgap = yes) [boolean]
Limit delta attitude interpolation if gaps present ([yes], no)

(attext = ATTITUDE) [string]
Attitude extension

(attcol = QPARAM) [string]
Attitude column

(attform = QUAT) [string]
Attitude format ([QUAT], EULER)

(orbext = ORBIT) [string]
Orbit extension

(orbcol = VELOCITY) [string]
Orbital velocity column

(orbform = VECTOR) [string]
Orbital velocity format ([VECTOR], COMPONENTS, KEPLERIAN)

(coordevt_randomize = TELDEF) [string]
Randomize coordinates when rebinning ([TELDEF], yes, no)

(randsys = TELDEF) [string]
Starting system for randomization (or TELDEF)

(randscalesys = TELDEF) [string]
System to determine randomization amount (or TELDEF)

(infileext = EVENTS) [string]
Event extension

(inclfloatcol = no) [boolean]
Write non-rounded coordinate columns (yes, [no])

(inclfloatskycol = no) [boolean]
Write non-rounded sky coordinate columns (yes, [no])

(floatcolsuffix = _FLOAT) [string]
Suffix for non-rounded coordinate columns

(startwithfloat = no) [boolean]
Start with non-rounded startsys coordinates (yes, [no])

(blankcol = yes) [boolean]
Assign null values to columns not calculated ([yes], no)

(btnull = 255) [int]
TNULL for byte (B) columns

(itnull = -999) [int]
TNULL for short (I) columns

(jtnull = -999) [int]
TNULL for long (J) columns

(ktnull = -999) [int]
TNULL for long (K) columns

(sbtnull = 255) [int]
TNULL for signed byte columns

(uitnull = -999) [int]
TNULL for unsigned short columns

(ujtnull = -999) [int]
TNULL for unsigned long columns

(colbound = -32768) [string]
TNULL, TLMIN, TLMAX for PHAS

(chipcol = CCD_ID) [string]
Chip column (or NONE)

(xcol = ACTX) [string]
X coordinate column

(ycol = ACTY) [string]
Y coordinate column

(chancol = PI) [string]
Pulse height column (or NONE)

(gradecol = GRADE) [string]
Event grade column (or NONE)

(grade = 0) [string]
Event grade for clean (or ALL)

(n_division = 1) [int]
Divide total observation time into the given number

(cleanimg = no) [boolean]
Output cleaned image for debugging (yes, no)

(cellsize = 7) [int]
Poisson clean cell size (odd integer > 1)

(impfac = 320) [double]
Factor for gamma function

(logprob1 = -5.6) [double]
Log Poisson probability threshold

(logprob2 = -5.6) [double]
Log Poisson probability threshold for second step

(iterate = yes) [boolean]
Iterate the second step Poisson clean (yes, no)

(flagedge = no) [boolean]
Zero chip edge pixels (yes, no)

(bthresh = 3) [int]
Zero background threshold

(duration = no) [boolean]
Perform detailed search for flickering duration (yes, no)

(sigma = 3.0) [double]
Significance level for flickering duration

(firstchip = TLMIN) [string]
Min value for chip number

(lastchip = TLMAX) [string]
Max value for chip number

(xmin = TLMIN) [string]
Min value for X coordinate

(xmax = TLMAX) [string]
Max value for X coordinate

(ymin = TLMIN) [string]
Min value for Y coordinate

(ymax = TLMAX) [string]
Max value for Y coordinate

(chanmin = TLMIN) [string]
Min pulse-height value for clean (inclusive)

(chanmax = TLMAX) [string]
Max pulse-height value for clean (inclusive)

(outbadpix = no) [boolean]
Output bad pixel file (yes/[no]). This parameter is not a boolean in sxiflagpix but rather a filename. For ahpipeline and sxipipeline this is boolean to account for multiple files

(outbadimg = yes) [boolean]
Output bad pixel image ([yes]/no). This parameter is not a boolean in sxiflagpix but rather a filename. For ahpipeline and sxipipeline this is boolean to account for multiple files

(npixnbr = 1) [int]
Pixel distance defining a neighbor

(nboundnbr = 1) [int]
Pixel distance defining neighbor from CCD/window/segment boundary

(citrailnbr = 2) [int]
Pixel distance trailing CI row

(ciprenbr = 1) [int]
Pixel distance preceding CI row

(echoflag = yes) [boolean]
If set to yes, cosmic ray echo pixels are flagged ([yes]/no).

(echomap = no) [boolean]
Output CR echo pixel fraction map (yes/[no])

(echonbr = 2) [integer]
Distance in pixels from a cosmic ray echo pixel to flag a neighbor pixel

(echomin = 6) [integer]
Minimum number of events for the cosmic ray echo fraction calculation

(echospth = 15) [integer]
Split threshold for cosmic ray echo fraction calculation

(echofrac = 0.7) [float]
Minimum fraction of hits defining a cosmic ray echo pixel. For any pixel contained in at least 'echomin' events, if at least 'echofrac' of those events have a pulse height above 'echospth', then the pixel is considered a cosmic ray echo pixel. If 'echoflag=no', this parameter is ignored

(bad_status = 3:9,11,12,16:19,25:28,30) [string]
Bad status list, colons can used to specify a range (e.g. 1:3,5 = 1,2,3,5)

(copyphas = yes) [boolean]
Copy original PHAS before processing ([yes]/no)

(resetflags = yes) [boolean]
Reset all sxiflagpix STATUS flags ([yes]/no)

(hkext = HK_SXI_USR_USER_HK1) [string]
HK extension with video temperatures

(hkcolstem = SXI_USR_HKTBL_) [string]
Column name stem for video temperatures

(hkvideoid = A,B,B,B) [string]
Video card ID for gain correction of CCD1-4

(startcol = PHAS) [string]
Starting point of correction

(evnoddcor = yes) [boolean]
Enable even-odd correction [yes/no]

(chtrailcor = yes) [boolean]
Enable charge trail correction [yes/no]

(cticor = yes) [boolean]
Enable CTI correction [yes/no]

(gaincor = yes) [boolean]
Enable gain correction [yes/no]

(ctigrade = no) [boolean]
Use grade information in CTI correction [yes/no]

(copygrade = no) [boolean]
Copy existing GRADE and PHA columns [yes/no]

(phcut = CALDB) [string]
Pulseheight cut for CTI correction, or CALDB

(badpixopt = 2) [int]
Options for events with bad pixels: ignore bad pixels (1), null bad pixels (2), null whole event (3)

(spthiter = yes) [boolean]
Enable split threshold iteration [yes/no]

(spthcaldb = yes) [boolean]
Use split thresholds from spthfile [yes/no]

(spthoffset = 15.) [real]
Split threshold offset value (if spthcaldb = no)

(spthslope = 0.) [real]
Split threshold slope value (if spthcaldb = no)

(evtthre = DEFAULT) [string]
Event threshold (or DEFAULT)

(negthre = -5000) [int]
Minimum PHAS value for normal event

(deltatime = 8) [int]
Max allowed time gap in HK temp search [s]

(debugcol = no) [boolean]
Write out the debug columns [yes/no]

(randomize = yes) [boolean]
Allow randomization in PI to UPI conversion (yes, no)

(timecol = TIME) [string]
Time column

(seed = 0) [int]
Random number generator seed (0=use system time)

(stemreport =) [string]
File stem for log and temporary files. If the parameter is not set the script automatically sets the stem to "sxipipeline_YYYYMMDDTHHMMSS_" and appends log file and temp file names as needed. Intended to be set by ahpipeline.

(numerrs = 0) [string]
Number of errors from sxipipeline (output)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
1. The following command recalibrates (stage 1) and re-screens (stage 2) all SXI data for sequence 100039010 that currently resides in the directory /data/100039010/sxi/event_uf, and the output is stored in a directory called /data/100039010_reproc/:
sxipipeline indir=/data/100039010/sxi/event_uf outdir=/data/100039010_reproc entry_stage=1 exit_stage=2 steminputs=ah100039010 \
attitude=/data/100039010/auxil/ah100039010/ah100039010.att orbit=/data/100039010/auxil/ah100039010/ah100039010.orb \
obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti housekeeping=/data/100039010/sxi/hk/ah100039010sxi_a0.hk \
makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \

extended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk

2. The following command re-screens (stage 2 only) SXI data for the same data set as in the previous example, as well as recalculate the SXI mode GTI:
sxipipeline indir=/data/100039010/sxi/event_uf outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 steminputs=ah100039010 \
calc_modegti=yes obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti \
makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk

3. The following command creates products (stage 3 only) SXI data for a calibrated data set:
sxipipeline indir=/data/100039010/sxi/event_cl outdir=/data/100039010_reproc entry_stage=2 \
exit_stage=2 steminputs=ah100039010 regionfile=none

NOTES
None, but see help for individual parameters above.

NAME sxirmf -- Create an SXI energy Redistribution Matrix File (RMF)

USAGE sxirmf infile outfile

DESCRIPTION
'sxirmf' creates an SXI RMF file that may be used for spectral analysis with an input spectrum file. The output RMF contains a matrix that, for a given incoming X-ray photon energy, describes the normalized probability of detecting that photon in any PI (pulse invariant) channel. This photon can be redistributed to PI channels away from the expected energy due to interactions with different material in the detector. For the SXI, the response is composed of: (1) a primary Gaussian component, centered at the expected energy, for photons that interact and are detected completely in the sensitive layer of the detector; (2) a secondary Gaussian sub-peak at slightly lower energy, for photons that lose some charge to pixels below the split threshold; (3) a Gaussian silicon escape peak centered at 1.74 keV below the primary peak, for photons that have some energy absorbed by a CCD Si atom; (4) a Gaussian silicon fluorescence peak at constant energy 1.74 keV, if the Si atom excited in the interaction with the X-ray photon emits a Si K alpha photon that is then detected in a different region; and (5) a constant continuum component extending from the peak energy to 0 eV, produced when a photon interacts near the insensitive detector layer and some charge is lost.

The SXI response has been measured on the ground using sources of monochromatic X-rays at a number of energies. However, increasing charge transfer inefficiency (CTI) due to radiation damage changes the response, especially the width of the Gaussian features, and so the RMF depends on both observation time and position on the detector. This information is extracted from the header keywords and the weighted map (WMAP) image, respectively, in the input spectrum file. The WMAP is a small counts image of the extracted events, which is used to weight the response parameters in each region before combining to produce a weighted response. Response parameters are defined in the CALDB at a number of energies, and interpolation is performed to place them on the energy grid in the MATRIX extension before the response is calculated.
The output RMF contains a MATRIX extension, which contains the response matrix; and an EBOUNDS extension listing the energy boundaries of each PI channel.
The energy bin size(s) of the RMF may be set with the 'dein' parameter, with 'eminin' and 'nchanin' determining the energy range for those bins (see below for more detail). The recommended energy bin sizes are 2 eV and 24 eV for the energies below and above 12 keV, respectively. The default parameter values of 'eminin=200', 'dein=2,24', and 'nchanin=5900,500' are determined so that these recommended MATRIX bins are calculated. Furthermore, the grid energy of 'EBOUNDS' must match the PI grid of the spectrum file, so the default parameter values of 'deout=6', 'eminout=0', and 'nchanout=4096' should be maintained, unless users want to change them for their specific purpose. Changing any of the hidden parameter values for this tool is not recommended and must be done with care.

PARAMETERS
infile [filename]
Name of input spectrum (PHA) file. It must contains a WMAP image rendered in the DET coordinate system in the primary extension.

outfile [filename]
Name of output response (RMF) file

(rmfparam = CALDB) [filename]
Name of input file from which the RMF parameters are read

(eminin = 200.0) [real]
Minimum energy of the MATRIX grid in eV.

(dein = 2,24) [string]
Spacing of energy bins (in eV) in each interval of the MATRIX grid. A single value (e.g., 2) applies a constant grid spacing for the entire matrix. Multiple values separated by commas (e.g., '2,24' as the default) create a variable grid spacing starting at 'eminin'. The number of channels in each energy interval must be specified with 'nchanin'. The values can be floating point.

(nchanin = 5900,500) [string]
Number of channels in each interval of the MATRIX grid. Combined with the default values of the previous two parameters, the default of '5900,500' means that the output MATRIX has 5900 bins (with 2 eV size) between 200 eV and 12000 eV, and 500 bins (with 24 eV size) above 12000 eV (hence up to 24000 eV). 'dein' and 'nchanin' must have the same number of intervals. The values of 'nchanin' must be integers.

(eminout = 0.0) [real]
Minimum energy of EBOUND in eV. The default value is 0.

(deout = 6.0) [real]
Spacing of the EBOUND energy grid in eV. This must be consistent with the assumed energy grid of the PHA file (i.e., 6 eV for the SXI).

(nchanout = 4096) [integer]
Number of channels in EBOUND. This must be consistent with the number of PI channels in the PHA file. The default value for SXI is 4096.

(rmfthresh = 1.e-6) [real]
Low threshold for the RMF construction. For a given X-ray photon energy, the normalized probability (i.e., a value less than 1) of producing a certain PI value is calculated for each channel and written in the MATRIX extension. If the probability is less than this threshold, 0 (zero) is written for that element. Using the default value (1.0e-6) is secure enough for most cases.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Create an RMF file with default energy grid and other settings.
sxirmf infile="spectrum.pha" outfile="response.rmf"

2. Create an RMF file with energy grid of 2 eV for the entire energy band 0.2-24 keV.
sxirmf infile="spectrum.pha" outfile="response.rmf" dein=2 nchanin=11900

NAME sxsanticolc - Extract SXS antico light curve(s) and spectra using optional screening criteria

USAGE sxsanticolc infile outroot

DESCRIPTION

'sxsanticolc' is a script that takes an SXS antico event file and generates a light curve after optional cleaning. 'sxsanticolc' assumes that the input file has the PI column already calculated using the 'sxsanticopi' task.

The PSP can be selected using the pspid parameter. If pspid is not set to A or B, then all events are used.

Before calculating the light curve the event file is filtered for all GTI. Any input gtifile is merged with a GTI attached to the input event file as well as events between parameters lcstart and lcstop. The event file is then screened for events based on the expr parameter.

Once the event file has been filtered, the 'extractor' tool is run to create a light curve for each picut. The light curve contains bins between lcstart and lcstop. If lcstart or lcstop are set to -1 (default) then TSTART or TSTOP are used for the light curve start and stop times, respectively. Additional control parameters are lcthresh and lctzero. The columns RATE and ERROR are appended to the output lightcurve for each PI range with the range in the column name. For example, if the picut="0-12200,0-1200,1201-12200" then the RATE column names in the output file are: RATE0_12200, RATE0_1200 and RATE1201_12200.

Optionally, the extract parameter creates a spectrum using 'extractor' and the full PI range for antico events (PI=0:12200).
PARAMETERS

infile [file]
   Input event file

outroot [string]
   Root of output file names

(extract = yes) [boolean]
   Extract spectrum ([yes],no)

(bintime = 1.0) [real]
   Time bin in seconds (d/f= 1 sec)

(antpsp = A) [string]
   Antico PSP to use in light curve (A=PSPA B=PSPB or NONE)

(expr = FLG_BASELINE==b0&&PI>=0&&PI<=12200&&DURATION>2&&DURATION<19) [string]
   Expression for event columns selection (or NONE)

(gtifile = NONE) [string]
   Input GTI file

(numlc = 1) [int]
   Number of lightcurve to output (d/f=1)

(picut = 60-12200) [string]
   PI ranges to create the lightcurve(s) (ex. 0-200,201-1000, etc.)

(lcstart = -1) [real]
   Force the lightcurve to start at specific MET time (if -1 use the 1st start GTI)

(lcstop = -1) [real]
   Force the lightcurve to stop at specific MET time (if -1 use the last stop GTI)

(lcthresh = 1) [real]
   Lightcurve exposure threshold

(lctzero = no) [boolean]
   Set the TIMEZERO to the first bin of the lightcurve (yes/[no])

(cleanup = yes) [boolean]
   Delete temporary files ([yes]/no)

(clobber = no) [boolean]
   Overwrite existing output file ([yes]/no)

(chatter = 2) [int]
   Chatter level for output

(logfile = !DEFAULT) [string]
   Output log file (DEFAULT, NONE; '!' to clobber)

(debug = no) [boolean]
   Enable debug mode ([yes]/no)

(history = yes) [boolean]
   Record tool parameters in HISTORY ([yes]/no)

(mode = ql) [string]
   Mode of automatic parameters

[ccldhm]

Mode of automatic PARAMETERS

EXAMPLES
1. Extract a lightcurve (sxsanticolc.lc) and spectrum (sxsanticolc.pha) using PSP A, specified GTI file and expression. Use a 20 second time bin with three energy ranges.
sxsanticolc "infile=ah100050050sxs_a0ac_uf.evt" "outroot=sxsanticolc" "bintime=20" "antpsp=A" \
"picut=0-12200,0-1200,1201-12200" "expr=FLG_BASELINE==b0&&PI>=0&&PI2&&DURATION<19" \
"gtifile=input/ah100050050sxs_el.gti[4]" "numlc=3"

2. Extract a lightcurve using PSP A with a 1 second bin time and constricting the lightcurve between lcstart and lcstop. The default energy range is used (0-12200)
sxsanticolc "infile=input/ah100050050sxs_a0ac_uf.evt" "bintime=1" "expr=NONE" \
"outroot=output/sxsanticolc" "antpsp=A" "lcstart=70203712" "lcstop=70205402"

NAME sxsanticopi -- Assign PHA and PI columns to the SXS anticoincidence events

USAGE sxsanticopi infile outfile gainfile

DESCRIPTION
'sxsanticopi' calculates the PHA and PI values of the events detected by the SXS anticoincidence. The PHA is calculated by taking the difference of the values in the ADC_SAMPLE_MAX and ADC_SAMPLE_PEDESTAL columns present in the antico event data file. To calculate the PI, 'sxsanticopi' uses the coefficient of a polynomial stored in the antico gain CALDB file for each of the PSP. The same antico events are read by both PSP-A and PSP-B electronics and stored in the event file. The column PSP_ID has the telemetered information of the PSP value. To calculate the PI, 'sxsanticopi' uses the coefficients of a polynomial stored in the antico gain CALDB file for both PSP. For each antico event, 'sxsanticopi' first reads the PSP_ID value and after applies the gain coefficients corresponding to the appropriate PSP. If the event is a 'LOST' events the PHA and PI are set to TNULL values (-32768). The number of PI channels are 12200 and each channel corresponds to 0.5 keV bin.

PARAMETERS
infile [filename]
Input antico event file name.

outfile [filename]
Output antico event file name.

gainantfile [filename]
Name of the calibration file with the polynomial coefficients to convert PHA to PI for both PSP-A and PSP-B or CALDB (default).

(acphaoffset = 0.0) [float]
Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+acphaoffset and +0.5+acphaoffset. So, when acphaoffset=0.5, the random offset is between 0 and 1.

(randomize = yes) [boolean]
If set to yes, PI is calculated from PHA using decimal randomization ([yes]/no).

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate the antico PHA, and the antico PI for events in antico_in.fits. Use the gain coefficients in CALDB and randomization to calculate the PI values, and include these in the output file antico_out.fits.
sxsanticopi antico_in.fits antico_out.fits CALDB

NAME sxsbranch -- Compute rates, branching ratios, and effective exposure times for each SXS event grade for an input real or simulated file, or based on a count rate

USAGE sxsbranch infile filetype outfile pixfrac pixmask countrate exposure

DESCRIPTION
'sxsbranch' computes rates and branching ratios for each SXS event grade based on the inputs (file or count rate). These are calculated for each pixel and for the entire array. 'sxsbranch' also statistically estimates the same quantities using the total rates from the input (file or rate) for the entire array using Poisson statistics. 'sxsbranch' operates in three modes: (1) no input file, (2) input simulated file, and (3) input file from observations.

(1) If there is no input file ('infile=NONE'), 'sxsbranch' internally simulates events using the 'exposure' and 'countrate' parameters, with counts distributed in the array according to the file specified by the 'pixfrac' parameter. If 'pixfrac=NONE' a uniform illumination is assumed. Times are randomly assigned, and event grades assigned accordingly. The branching ratio is calculated using these simulated events. There are two different supported timing distributions within the exposure: constant count rate profile ('flagburst=no') and burst-like profile ('flagburst=yes'). To simulate a burst-like lightcurve within the exposure, the following parameters also must be set: 'ratburst' is the ratio of burst-peak/quiescent count rate; 'tburst' gives the burst start time in sec after the beginning of the exposure; 'trise' is the rise time (assumed linear) in sec; and 'tdecay' is the decay time (assumed exponential) in sec. The grades are assigned by comparing the resulting time intervals between events with the time interval values in the parameters 'dtmidhigh', 'dtlowmid', and 'dtprimary': (a) high-resolution events have no preceding or following events within 'dtmidhigh'; (b) mid-resolution events have at least one preceding or following event within 'dtmidhigh', but none within 'dtlowmid'; (c) low-resolution events at least one preceding or following event within 'dtlowmid'. (d) Primary events are defined as having no preceding event within 'dtprimary'; (e) secondary events at least one preceding event within 'dtprimary'. The outer 20 pixels may be excluded from the branching ratio calculations by setting the parameter 'pixmask=pixmask.txt'.

(2) If the input file is an SXS event file from 'heasim' (infile set to the filename of the SXS event output file simulated with 'heasim' and 'filetype=sim'), 'sxsbranch' calculates the branching ratio using the simulated SXS file. The simulated files do not have grades assigned by 'heasim'. 'sxsbranch' first considers if events are crosstalking (the option if the parameters 'ctphafrac1' and/or 'ctphafrac2' are different from zero), and subsequently assigns the grade using the event time and the calculated time intervals to the preceding and following events in the same pixel. The parameters used to calculate crosstalk are 'debin', 'enrgthr', 'ctphafrac1', and 'ctphafrac2'. These parameters are applied to all pixels. There are two crosstalk types: (a) the first occurs when an event induces crosstalk in the nearest electrical bonded pixel, (b) the second occurs when an event induces crosstalk in the next-nearest electrical bonded pixel. The first is calculated if 'ctphafrac1' is different from 0 and the event has ctphafrac1*PI > (enrgthr/debin). The second is calculated if 'ctphafrac2 'is different from 0 and the event has ctphafrac2*PI > (enrgthr /debin). The parameters 'ctphafrac1' and 'ctphafrac2' define the fraction of crosstalk energy to the original event energy, 'enrgthr' is the energy threshold above which crosstalk triggers, and 'debin' is the value of the PI bin in eV. The maximum possible crosstalk is induced for the default 'enrgthr=0'. The electrical proximity of the pixels is defined by the calibration map pixmap.fits (set by the parameter 'ctelpixfile'). Grades are calculated using the 'dtmidhigh', 'dtlowmid', and 'dtprimary' parameters as described above. The outer 20 pixels may be excluded from the branching ratio calculations by setting the parameter 'pixmask=pixmask.txt'.

3) If the input SXS event file is from an observation (infile set to the SXS event filename from an observation and 'filetype=real'), 'sxsbranch' calculates the branching ratio using this SXS observation event file. This option may be used after launch since the branching ratio calculation uses the on-board assigned grades. Baseline and lost events are excluded from the branching ratio calculation. The outer 20 pixels may be excluded from the branching ratios calculation by setting the parameter 'pixmask=pixmask.txt'.

For comparison with the grade distribution statistics compiled from real or simulated events, statistical estimates are computed and output. For these, the count rate is distributed on the array according to the file specified by the 'pixfrac' parameter. If 'pixfrac=NONE' a uniform illumination is assumed. For pixel 12, the calpixel, the rate is set to the value specified in the 'calpixrate' parameter. If 'ctphafrac1' and/or 'ctphafrac2' are > 0 crosstalk is calculated assuming that any pixel crosstalks with all of its nearest and/or next-nearest pixels as defined by the electrical proximity calibration map pixmap.fits (set by the parameter 'ctelpixfile'). The outer 20 pixels may be excluded from the branching ratios calculation by setting the parameter 'pixmask=pixmask.txt'. The fraction of different grades in each pixel is assigned using the 'dtmidhigh', 'dtlowmid', and 'dtprimary' parameters as described above, and assuming Poisson statistics.

'sxsbranch' can simulate pileup defined as occurring when events in a given pixel separated by small intervals cannot be distinguished, if the 'flagmerge' parameter is set to yes. This is valid either for simulated ('heasim') event input files, or if 'infile=NONE'. In this case, events separated in time by intervals smaller than that given by the 'dtmerge' parameter are combined into a single event up to a total time interval given by the 'dtmax' parameter. If there is a 'heasim' input file, the energy of the merged event is set equal to the sum of energies of the individual events that were merged, and the merged event is discarded if the value exceeds that given by the 'maxenergy' parameter. Crosstalk events are added, and events are graded based on the event list after merging.

(a) The pixfrac.txt file is used in the statistical estimates, and for simulating events if 'infile=none'. The file contains the distribution fraction of the counts in the total array based on the PSF. (b) The pixmap.fits file is used when including crosstalk in calculating branching ratios for 'filetype=sim' or 'infile=none', as well as for the associated statistical estimates. The file contains information on the electrical proximity of the pixels necessary to define crosstalk among pixels. (c) The pixmask.txt is used to exclude pixels from the calculation when calculating branching ratios, and for the associated statistical estimates.

The output of 'sxsbranch' is a FITS file with two extensions with identical structure. The first extension, BRANCHEST, contains the results of the statistical estimates based on Poisson statistics. The second extension, BRANCHCALC, contains the results of calculations based on the input data file or count rate. Each extension contains a group of keywords providing the results for the full array and columns providing the results for each pixel. The total rates are obtained considering the sum of good and crosstalk events. If the file is from an observation, rather than a simulation, the good events do not include baseline or lost events. If the file is from a simulation, the good events do not include crosstalk. The rates and branching ratios per grade are averages over the entire observation. If the input file is a simulated file from 'heasim', 'sxsbranch' creates a copy of the input file, adding and filling two columns: PIXEL with the pixel number, and ITYPE with the grade (ITYPE = 0, 1, 2, 3, 4 for HP, MP, MS, LP, LS, respectively). Additionally, if 'flagmerge=yes' the PILEUP column in the output events file is set to 1 for merged events and 0 otherwise.

PARAMETERS

infile [file]
Name of the input event FITS file. This file is either the output event file from a heasim SXS simulation, or an event file from an observation. Infile may be set to NONE when the branching ratio is calculated using the value input in the parameters countrate and exposure. If filetype is set to sim, an event file with added PIXEL and ITYPE columns is created with this name and ".out" appended.

filetype [string]
Type of the input event file. Set filetype to sim if the infile is a simulated event file from heasim; set filetype to real if the infile is from an observation. This parameter is not used if infile is set to NONE.

outfile [file]
Name of the output sxsbranch file with the branching ratio calculation.

countrate [real]
Count rate [ct/s] for the entire array. This parameter is only valid if infile is set to NONE.

exposure [real]
Exposure time [s] for the entire array. This parameter is only valid if infile is set to NONE.

pixfrac [string]
Name of the ASCII file with the distribution of the fractional counts in the pixels. The file has two columns. The first contains the SXS pixel number. The second contains the fraction of counts that fall in each pixel. If set to NONE, the count distribution is uniform in all pixels. This is used for the statistical estimates, and to distribute counts when infile is set to NONE.

pixmask [string]
Name of the ASCII file to exclude pixels in all calculations of the branching ratio. The file has two columns. The first contains the SXS pixel number. The second is set either to 0 or 1 to include or exclude pixels in the calculation, respectively. If pixmask is set to NONE all pixels are included. This parameter is valid for all calculations, as well as for the associated statistical estimates.

(ctelpixfile = CALDB) [file]
Name of the FITS file with the SXS electrical pixel map. The file contains the mapping of how the pixels are wired and therefore the pixels that are nearest and next nearest when considering crosstalk. If the parameter is set to CALDB, the file is read from the calibration database.

(dtprimary = 69.92) [string]
Time interval [ms] used to distinguish primary from secondary events in the grade calculation. The value is set to 69.92 ms and corresponds to the current on-board software setting. It should not be modified. If the parameter is set to CALDB, the file is read from the calibration database.

(dtmidhigh = 69.92) [string]
Time interval [ms] used to distinguish mid- from high-res. The value is set to 69.92 ms and corresponds to the current on-board software setting. It should not be modified. If the parameter is set to CALDB, the file is read from the calibration database.

(dtlowmid = 17.95) [string]

Time interval [ms] used to distinguish low- from mid-res. The value is set to 17.95 ms and corresponds to the current on-board software setting. It should not be modified. If the parameter is set to CALDB, the file is read from the calibration database.

(calpixrate = 6.0) [real]
Count rate in [cts/s] in the SXS calibration pixel (pixel 12). This parameter is only used for statistical estimates. The default value is 6.0 count/sec.

(ctphafrac1 = 0.006) [real]
Energy ratio of the nearest electrically bonded pixels to the original, due to crosstalk. This parameter is used for statistical estimates, and if infile is set to NONE or filetype is set to sim. For the latter, events of energy E are assumed to induce crosstalk in nearest electrically bonded pixels, if ctphafrac1 * E > enrgthr. For the statistical estimate of branching ratios or if infile is set to NONE, this crosstalk is considered if ctphafrac1 > 0, since energy is irrelevant to the procedure.

(ctphafrac2 = 0.001) [real]
Energy ratio of the next-nearest electrically bonded pixels to the original, due to crosstalk. This parameter is used for statistical estimates, or if infile is set to NONE or filetype is set to sim. For the latter, events of energy E are assumed to induce crosstalk in nearest electrically bonded pixels, if ctphafrac2 * E > enrgthr. For the statistical estimate of branching ratios or if infile is set to NONE, this crosstalk is considered if ctphafrac2 > 0, since energy is irrelevant to the procedure.

(debin = 0.5) [real]
Value in eV to convert a PI channel bin to energy. This parameter is only valid if filetype is set to sim. The conversion of PI to energy is used in the crosstalk calculation. The current value for the output of heasim should be set to 1 eV.

(enrgthr = 0) [real]
Energy threshold, in eV. This parameter is only valid if filetype is set to sim. Events of energy E are assumed to induce crosstalk in adjacent pixels, if ctphafrac1 * E > enrgthr, and next-nearest pixels if ctphafrac2 * E > enrgthr.

(flagmerge = no) [boolean]
Flag to merge events separated in time by intervals smaller than that given by the dtmerge parameter. This parameter is valid if infile is set to NONE or if filetype is set to sim. Consecutive events separated in time by less than dtmerge in a given pixel are combined into a single event up to a total time interval given by the dtmax parameter, and discarded if the sum of their energies exceeds the value given by the maxenergy parameter. Crosstalk events are added, and events are graded based on the event list after merging. If filetype is set to sim, the PILEUP column in the output events file is set to 1 for such merged events and 0 otherwise (yes/[no]).

(dtmerge= 2.0) [real]
Time interval [ms] within which consecutive events are merged. This parameter is valid if infile is set to NONE or if filetype is set to sim, and flagmerge is set to yes. Must be less than dtlowmid, and reflect the limitations of the on-board electronics to distinguish events within dtmerge.

(dtmax = 12.0) [real]
Maximum time [ms] within which consecutive events are merged. This parameter is valid if infile is set to NONE or if filetype is sim, and flagmerge is set to yes. If events arrive separated in time by less than the interval defined by dtmerge, dtmax defines the maximum total time interval within which events are merged.

(maxenergy = 15.0) [real]
Maximum energy, in keV, for merged events. Merged events created from events with total summed energy greater than this are discarded. This parameter is valid if infile is set to NONE or if filetype is set to sim, and flagmerge is set to yes.

(flagburst = no) [boolean]
Flag to distribute the count rate as a burst-like lightcurve. This parameter is only valid for infile is set to NONE. If set to yes, event times are distributed as a burst profile lightcurve defined by the parameters ratburst, tburst, trise, and tdecay. If set to no, the count distribution is uniform within the exposure (yes/[no]).

(ratburst = 30.0) [real]
Ratio of the burst peak rate to the quiescent rate. This parameter is only valid if infile is set to NONE and flagburst is set to yes.

(tburst = 10.0) [real]
Start time of the burst in sec after the beginning of the exposure. This parameter is only valid if infile is set to NONE and flagburst is set to yes.

(trise = 3.0) [real]

Rise time in sec for the burst profile. The rise time is assumed to be linear. This parameter is only valid if infile is set to NONE and flagburst is set to yes.

(tdecay = 60.0) [real]
Decay time in sec for the burst profile. The decay time is assumed exponential. This parameter is only valid if infile is set to NONE and flagburst is set to yes.

(tstart = 0.0) [real]
Time since 2014 [s] to use for CALDB query if infile is set to NONE.

(dmmin = 0) [real]
Minimum deriv_max to select good events. This parameter is only valid if filetype is set to real. Events with derive_max

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows)
[ccldhm]

EXAMPLES
1.Calculate the branching for the real sxs event file sxs_real.fits, writing the output to sxsbranch.out. For purposes of the semi-analytic estimate, distribute the source counts as a point source.
sxsbranch infile=sxs_real.fits filetype=real outfile=sxsbranch.out pixfrac=pixfrac.txt pixmask=NONE ctelpixfile=pixmap.fits

2. Calculate the branching for a simulated event file, ignoring crosstalk in the branching calculations and estimates, and the calpixel in the estimates. Mask the outer 20 pixels from the calculations and estimates.
sxsbranch infile=sim.fits filetype=sim outfile=sxsbranch.out pixfrac=pixfrac.txt\
pixmask=pixmask.txt ctelpixfile=pixmap.fits ctphafrac1=0.0 ctphafrac2=0.0 calpixrate=0.0

3. Calculate the branching for a simulated event file with 1 eV PI channel widths, including crosstalk in the branching calculations and estimates, and the calpixel in the estimates. For the simulated event list, crosstalk events are added to nearest-bonded pixels if 0.00125 X the original energy exceeds a 10 eV threshold, and to next-nearest-bonded pixels if 0.0005 X the original energy exceeds 10 eV.
sxsbranch infile=sim.fits filetype=sim outfile=outfile=sxsbranch.out pixfrac=pixfrac.txt pixmask=none ctelpixfile=pixmap.fits \
ctphafrac1=0.00125 ctphafrac2=0.0005 debin=1.0 enrgthr=0.0

4. Calculate the branching based on a constant count rate of 100 cts/sec and an exposure of 1 ksec (no input file), where the counts are distributed as a point source. Include all possible crosstalk in the branching calculations and estimates.
sxsbranch infile=NONE filetype=NONE outfile=sxsbranch.out 100 1000 ctphafrac2=0.0001 \
pixfrac=pixfrac.txt pixmask=none ctelpixfile=pixmap.fits ctphafrac1=0.001 \

5. Calculate the branching based on a count rate of 100 cts/sec and an exposure of 1 ksec. The counts are distributed as a point source, and follow a burst-like light curve with 1 second risetime, 10 second decay time, and burst-to-quiescent ratio of 100. The burst begins at the beginning of the "exposure".
sxsbranch infile=NONE filetype=NONE outfile=sxsbranch.out 100 1000 pixfrac=pixfrac.txt pixmask=none \
ctelpixfile=pixmap.fits ctphafrac1=0.001 ctphafrac2=0.0001 flagburst=yes ratburst=100.0 tburst=0.0 trise=1.0 tdecay=10.0

6. Calculate the branching for a simulated event file with 1 eV PI channel widths, merging events separated in time by less than 2 ms up to a maximum total interval of 30 ms, discarding events where the sum of the energies of the merged events exceed 15 keV.
sxsbranch infile=sim.fits filetype=sim outfile=sxsbranch.out pixfrac=pixfrac.txt pixmask=none ctelpixfile=pixmap.fits ctphafrac1=0.0 \
ctphafrac2=0.0 calpixrate=0.0 debin=1.0 flagmerge=yes dtmax=30.0 maxenergy=15.0

NAME sxsextend -- Calculate the extended PI for the SXS event files

USAGE sxsextend inuffile outuffile outclfile driftfile gtigenfile gtitelfile gtimxsfile

DESCRIPTION
The baseline PI column populated by the pipeline has a maximun number of channels of 32768. Each channel is 0.5 eV and the maximun energy in the spectrum is about 16.4 keV. The SXS may have signal collected up to 32 keV. 'sxsextend' allows to extend the range of channels and therefore the range of energy. The result is stored in a column named PIE.

'sxsextend' is a script that starting from the unfiltered event file run in sequence 'sxspha2pi', 'sxsperseus' (only for early Perseus data) and re-screen the data. Several files are required as input to 'sxsextend': the ufiltered event file, several gti files and the drift correction file. All these files are present in the archive. To extend the energy the parameters 'eminin', minimum energy in eV, 'dein', bin size in eV, and 'nchanin', number of channels, are set to default value to produce a spectra up to 32 keV with a bin size of 1 eV. The output are the unfiltered event file where the column PIE is populated and the cleaned event file obtained by screening the unfiltered data using the input GTI and event screening criteria stored in CALDB.

If the input event file has already the EPI column, it is possible to calculate the PIE by setting the parameter 'driftfile' to NONE and provide the value for the energy range in the 'eminin', 'offset' and 'nchanin'. This is a useful short cut when the driftfile is identical to that used to calculate the EPI column (see 'sxspi' and 'sxsgain').

PARAMETERS

inuffile [filename]
Input SXS event file name. This file maybe be the _uf.evt or the _cl.evt.

outuffile [filename]
Name of the output unfiltered event file with the column PIE populated. If the input is the _cl.evt the output is the _cl.evt with the PIE column populated.

outclfile [filename]
Name of the output cleaned event file with the column PIE populated. Event and GTI screening are applied. If the input file is the _cl.evt and not GTI are applied this file is identical to the 'outffile'.

driftfile [filename]
Input file containing drift correction file. This is the output of sxsgain. If set to NONE the code uses the input file and calcultes the PIE column from the EPI using the values in the parameters 'eminin', 'offset' and 'nchanin'.

gtigenfile [filename]
Input file containing the general gti. This file is archived with the data and has the suffix _gen.gti. If the paramater is set to NONE no tel GTI is used.

gtitelfile [filename]
Input file containing the good telemetry gti. This file is archived with the data and has the suffix _tel.gti If the paramater is set to NONE no tel GTI is used.

gtimxsfile [filename]
Input file containing the gti to exclude the MXS (modulated X-ary source). This file is archived with the data and has the suffix _mxfn.gti. If the paramater is set to NONE no mxs GTI is used.

gtiadroff [filename]
Input file containing the gti to exclude the time when the ADR is on. This GTI is an extention of the unfiltered archived event file. If the paramater is set to DEFAULT the task read these GTI from the extension.

gtimkf [filename]
Input file containing the gti to include time when the housekeeping are within the expected values This GTI is an extention of the unfiltered archived event file. If the paramater is set to DEFAULT the task read these GTI from the extension.

gtiehk [filename]
Input file containing the gti to include time when the orbital parameters are within the expected values This GTI is an extention of the unfiltered archived event file. If the paramater is set to DEFAULT the task read these GTI from the extension.

gtiextra [filename]
Input file containing the user gti to exclude other unwanted times. If the paramater is set to NONE no extra GTI file is used.

eminin = 0.0 [double]
Minimum energy in eV. Default is set to 0.0 eV.

offset = 1.0 [double]
Offset for first PIE for extended energy range [eV]. Default is set to 1.0 eV

nchanin = 32768 [int]

Maximum number of PIE channel for extended energy range.Default is set to 32768.

(gainfile = CALDB) [filename]
Input file containing SXS energy scale (gain) coefficients, or CALDB.

(scalefile = CALDB) [filename]
Input file containing SXS gain adjustments for each individual pixels, or CALDB. This adjustment is only valid if the gain has been calculated using the pixel 12. There are two allowable formats for the scalefile. The first format has two columns, PIXEL and HP, and the scale factors are applied to the event grades as specified in the scalegrade parameter. The second format has three columns, PIXEL, HP, M, and L, giving the specific scale factors for each grade. The scalegrade parameter is ignored when the task is given a scalefile with the second format.

(dgfile = REFDATA) [file]
Input gain coefficients file. This file is read from the REFDATA area of HEAsoft and should not be changed.

(offsetfile = REFDATA) [file]
Input calibration offset file. This file is read from the REFDATA area of HEAsoft and should not be changed.

(selectfile = CALDB) [file]
Name of the select file or user input label file with labels and expressions. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information.

(leapsecfile = REFDATA) [file]
Name of the input leap second file. If set to CALDB or REFDATA uses the leapsecond file in either CALDB or REFDATA area.

(secphacol= PHA) [string]
Input column name to use for the gain correction. The default is PHA containing the telemetered values. If the secondary were corrected (see task sxsseccor) a new PHA column is present in the file (default PHA2) and the user may specified this new column name to apply the gain.

(method = FIT) [string]
Correction method (FIT or AVERAGE). If FIT, the TEMP_FIT column in the driftfile is used; if AVERAGE the TEMP_AVG column is used.

(scaleepi= no) [boolean]
Scale EPI for each pixel using the values in the CALDB file entered in the parameter scalefile (yes/[no]).

(scalegrade= 0) [string]
Grade to which the scaling factor per pixel is applied. By default the task applies the scaling factor only to grade 0. Other grades maybe be specified by a comma separated list, e.g. "0,1,2". This parameter is only used when the scalefile contains a single column (HP).

(itypecol = ITYPE) [string]
Column containing event grade in infile.

(gapdt = -1.) [double]
Time [s] between events to define a gap (or <0). Events must have entries appropriate to its pixel value in driftfile within this interval; otherwise, EPI and PI are set to their null values.

(ntemp = 3) [int]
Number of temperatures from gain file to use in interpolation. The energy at each of these temperatures is calculated from the PHA of the event, and used to derive the event energy by ntemp-point interpolation.

(writetemp = no) [boolean]
Write temperature used for each event to output event file (yes/[no]).

(extrap = no) [boolean]
Allow extrapolation when determining drift temperature (yes/[no]).

(randomize = yes) [boolean]
If randomize = 'yes', decimal randomization is applied to PHA (an integer) before applying the gain to obtain energy (a double).

(seed = 0) [integer]

Random number generator seed; uses system time for seed=0.

(outrange = NULL) [string]
Define how to handle events with TIMEs outside of the driftfile TIME range. If outrange = NULL, then assign EPI=PI=NULL; if outrange=CONST, then use the gain correction from the first or last time in driftfile; if outrange=EXTRAP, then extrapolate the gain correction from driftfile.

(expr [string])
User-input expression applied to input file used to create GTI.

(label = NONE) [string]
Labels to read from label file specified by 'selectfile'. To input multiple labels, use a comma-separated list.

[ccldhm]

EXAMPLES

1. Assign PIE for events in event_uf.fits and screen the data

```
sxsextend  event_uf.fits event_uf_out.fits event_cl_out.fits driftfile gti_gen.gti gti_tel.gti
gtimxsfile=NONE eminin=0.0 dein=1 nchain=32768
```

NAME sxsflagpix -- Flag SXS events for antico and MXS event coincidence, temporal proximity, and crosstalk

USAGE sxsflagpix infile inantfile outfile antpsp inmxsGTI

DESCRIPTION
'sxsflagpix' identifies SXS events for coincidence with antico detector events and mxs events, temporal proximity between two events in any pixel, and electrical and recoil crosstalk. The identifications are written in the STATUS column. 'sxsflagpix' excludes lost and baseline pixel events, and events with risetime exceeding 127 if 'ckrisetime=yes', when determining event-to-event coincidence for temporal proximity, electrical and recoil crosstalk. It also excludes events below a PI threshold, set in the parameter 'pxpithr', for any coincidence defined in the parameter 'usepxpithr'.
Any combination of flagging may be enabled by setting the 'calcant' (antico), 'calcprox' (proximity), 'calcctrec' (recoil crosstalk), 'calcctel' (electrical crosstalk), 'calctel2' (electrical crosstalk 2), and 'calcmxs' (MXS) switch parameters. If the parameter 'dtflag=yes', the time intervals between crosstalking events, or between pixel events and their coincident antico events, are written out to additional columns (DTCEL for electrical crosstalk, DTCEL2 for electrical crosstalk 2, DTCTREC for recoil crosstalk, and DTANT for antico coincidence) if the corresponding flagging is enabled. An antico event file is required for antico flagging (see parameter 'inantfile'). A GTI containing the times when the MXS is on is used to determine the MXS flagging (see parameter 'inmxsgti'). There is an option to input an additional GTI file, unrelated to the MXS, to flag events within that GTI (see 'gtifile' parameter). This file may contain multiple GTI extensions, one applicable to all pixels and the remaining applicable to each individual pixel.

The STATUS column has 16 bits, 14 of which are currently in use and two of which are spare. The first bit flags events that are out or invalid (0) or in and valid (1) of the GTI file applicable to all pixels. The second bit flags the events that are out or invalid (0) or in and valid (1) of the GTI file applicable to individual pixels. The third bit flags events that are coincident with antico events (1). The fourth bit flags events that are within a temporal proximity (defined by parameter 'proxdt') of another event in any pixel (1). The fifth and sixth bits flag events for recoil crosstalk. The seventh and eight, and thirteenth and fourteenth, bits flag electrical crosstalk. The ninth through twelfth bits flag coincidence with MXS events.

Two formats are supported for the GTI file. The first format contains up to 37 extensions: 1 general GTI and 36 pixel-dependent GTI. The DETNAM keyword for the general GTI must be PIXEL and the DETNAM keyword values for the pixel-dependent GTIs are PIXnn where nn is the pixel number, e.g. PIX04 for pixel 4. In this format, all extensions must contain columns START and STOP which define the GTI. The second format may have up to two extensions: 1 general GTI with START and STOP columns and 1 pixel-dependent extension with columns START, STOP, and PIXEL. In the second format the DETNAM keyword is set to PIXEL for both extensions. if the input GTI has a single GTI extension with no DETNAM keyword defined, it is treated as a general GTI. If the events need to be filtered using multiple general or pixel-dependent GTI files, they must first be merged into a single GTI file using the 'sxspixgti'.

Antico flagging: 'sxsflagpix' flags a pixel event as coincident with and an antico event if the following condition is satisfied: antdtpre < t(pixel)-t(antico)+antshift < antdtfol. The value of the parameters 'antdtpre' and 'antdtfol' are time intervals that define the window of coincidence, and the antico time is adjusted by the parameter 'antshift' to account for the differences in time assignment of a pixel event

compared to an antico event. The antico event file contains redundant entries for events channeled through both sides of the PSP (A and B). 'sxsflagpix' uses only events from one side. This is specified as either A or B by the 'antpsp' parameter. Only antico events with duration and PHA greater then the threshold values determined by the 'antdurthr' and 'antphathr' parameters are considered in antico flagging.

Recoil crosstalk flagging: 'sxsflagpix' flags a pixel event if it is a recoil crosstalk event with a cal-pixel (pixel 12) MnK X-ray event. An event is considered as a recoil crosstalk event if its time is within the time interval 'ctrecdt' of a calpixel event and satisfies the following condition: ENERGY(EVENT) <= ENERGY(MnKa_low) or ENERGY(MnKa_high) < ENERGY(EVENT) < ENERGY(MnKb_high), where ENERGY(MnKa_low), ENERGY(MnKa_high), and ENERGY(MnKb_high) are set by the parameters 'kalow', 'kahigh', and 'kbeta'.

Electrical crosstalk flagging: 'sxsflagpix' flags electrical crosstalk across groups of pixels that are in wiring proximity, using two different time intervals (one short, 'cteldt' parameter, and one long, 'cteldt2' parameter). For each time scale, the flagging uses two bits. STATUS bits 7-8 are assigned to the short timescale, and bits 13-14 to the long timescale as follows: 00 - no crosstalk, 10 - multiple pixels active within that timescale, 11 - multiple pixels active within that timescale and this is the event with the largest PHA in the multiple-pixel crosstalk group. Two electrically crosstalking pixels must be in the same quadrant of the SXS detector, and must have pixel indices within the quadrant separated by no more than the number given by the 'ctelnear' (short timescale) or 'ctelnear2' (long timescale) parameters. These quadrant assignments and indices are stored in the CALDB pixel map (see 'pixdeffile' parameter). If one or both of the electrical crosstalk flagging options is enabled, the crosstalk multiplicity is recorded in the two-digit CTMULT (short timescale) and/or CTMULT2 (long timescale) columns as follows. The first digit gives the number of events in the crosstalk group, and the second digit the PHA rank of the event in the group where 1 is assigned to the event with the highest PHA.

MXS flagging: 'sxsflagpix' flags pixel events that are coincident with MXS events defined using the MXS GTI (see 'inmxsgti' parameter). Bits 9-10 are reserved for events coincidence with MXS direct mode, 11-12 are reserved for events coincidence with MXS indirect mode. Specifically, events with time within the MXS GTI set bits 10 and 12. Events within the interval set by the parameter 'mxsdt' (the MXS afterglow) set bits 9 and 11.

PARAMETERS
infile [filename]
Input SXS event file name. The file must have time assigned and include a STATUS column.

inantfile [filename]
Input antico event file name. The file must have time assigned and PHA calculated. Required if antico-flagging is switched on (calcant=yes). Otherwise can be set to NONE.

outfile [filename]
Name of the output event file with STATUS assigned or reassigned.

antpsp [string]
If antpsp = "A" antico events processed through PSP-A only are checked for antico coincidence; if antpsp = "B" antico events processed through PSP-B only are checked for antico coincidence.

(antshift = CALDB) [string]
Time offset in seconds between event time and central time for defining the window used for antico flagging. If set to CALDB, the parameter is read from the calibration database.

(gtifile = NONE) [filename]
Input GTI filename. Events with times outside of all of the intervals are flagged, for all valid SXS GTI extensions. The GTI may be applied to all pixels in the SXS array if the extension header includes the PIXEL DETNAM keyword. Pixel-specific GTI files are identified by the PIXNN DETNAM header keyword, where NN=00, 01... 35).

(calcant = yes) [boolean]
If calcant = yes, events coincident with antico events are flagged and an antico input file is required; if calcant = no this is skipped ([yes]/no).

(antdtpre = CALDB) [string]
Time interval in seconds preceding the time of an antico event (shifted by antshift) used to define the lower limit of the window used for antico flagging. If set to CALDB, the parameter is read from the calibration database.

(antdtfol = CALDB) [string]
Time interval in seconds following the time of an antico event (shifted by antshift) used to define the upper limit of the window used for antico flagging. If set to CALDB, the parameter is read from the calibration database.

(antswitch = 1) [integer]
If antswitch = 0, values in the antico DURATION column from inantfile are used in place of antdtfol; if 1 the antdtfol parameter is used. Expected to always be 1.

(antphathr = 1) [integer]
PHA threshold for antico flagging. Only antico events with PHA above this value are considered.

(antdurthr = 1) [integer]
Duration threshold for antico flagging. Only antico events with duration above this value are considered.

(calcctrec = yes) [boolean]
If calcctrec = yes, events are checked for recoil crosstalk flagging; if calcctrec = no this is skipped ([yes]/no).

(ctrecdt = CALDB) [string]
Time interval in seconds that defines event crosstalk due to cal-pixel recoil electrons. Events occurring within ctrecdt of a cal-pixel event are flagged if calcctrec = yes. If set to CALDB, the parameter is read from the calibration database.

(calcprox = yes) [boolean]
If calcprox = yes, events are checked for basic temporal proximity with any other event ([yes]/no).

(proxdt = CALDB) [string]
Time interval between events in seconds required for proximity crosstalk to be flagged. If set to CALDB, the parameter is read from the calibration database.

(calcctel = yes) [boolean]
If calcctel = yes, events are checked for electrical crosstalk flagging using the cteldt time interval; if calcctel = no this is skipped ([yes]/no).

(pixdeffile = CALDB) [filename]
Input file containing the SXS electrical pixel map. If set to CALDB, the file is read from the calibration database.

(cteldt = CALDB) [string]
Time interval between events in seconds required for electrical crosstalk to be flagged. If set to CALDB, the parameter is read from the calibration database.

(ctelnear = 1) [integer]
Number of nearby pixels to be checked for electrical crosstalk, where proximity is defined by pixdeffile and restricted to quadrants of the SXS detector.

(calcctel2 = no) [boolean]
If calcctel2 = yes, events are checked for electrical crosstalk flagging using the cteldt2 time interval; if calcctel2 = no this is skipped. (yes/[no])

(cteldt2 = CALDB) [string]
Time interval between events in seconds required for electrical crosstalk 2 to be flagged. If set to CALDB, the parameter is read from the calibration database.

(ctelnear2 = 1) [integer]
Number of nearby pixels to be checked for electrical crosstalk 2, where proximity is defined by pixdeffile and restricted to quadrants of the SXS detector.

pxpithr = 600) [integer]
Events with PI values below this threshold are excluded from flagging checks given by the usepxpithr parameter.

(usepxpithr = ALL) [string]
A comma-delimited list specifying which flagging types should use the pxpithr parameter for excluding events. Allowed values in the list are ALL, NONE, PROX (proximity), CTEL (electrical cross talk), CTEL2 (2nd electrical cross talk), and CTREC (recoil cross talk). Events that do not belog to the types specified in the list are excluded from flagging regardless on their PI value.

(calcmxs = yes) [boolean]

If calcmxs = yes, events are checked for coincidence with MXS events; if calcmxs = no this is skipped ([yes]/no).

inmxsgti = mxs.gti [filename]
Input MXS GTI filename providing the MXS start and stop times; required if calcmxs = yes. Events are flagged for coincidence with MXS operation when their times fall within any of the intervals in this file.

(mxsdt = CALDB) [string]
Time interval by which the MXS pulse stop times given in the MXS GTI are extended to account for MXS afterglow. If set to CALDB, the parameter is read from the calibration database.

(kalow = 5860) [real]
Input Mn K-alpha minimum PHA value. A cal-pixel recoil event with PHA less than this value has an additional flag set specifying that the cal-pixel recoil crosstalk energy condition is met.

(kahigh = 5930) [real]
Input Mn K-alpha maximum PHA value. A cal-pixel recoil event with PHA greater than this value (but less than kbeta) has an additional flag set specifying that the cal-pixel recoil crosstalk energy condition is met.

(kbeta = 6450) [real]
Input Mn K-beta PHA value. A cal-pixel recoil event with PHA less than this value (but greater than kahigh) has an additional flag set specifying that the cal-pixel recoil crosstalk energy condition is met.

(dtflag = no) [boolean]
Add delta-time columns for cross-talk and antico (yes/[no]).

(ckrisetime = yes) [boolean]
Do not flag, and do use for flagging, events with RISE_TIME > 127 ([yes]/no).

(resetflags = no) [boolean]
Reset all STATUS flags to zero before applying new flags (yes/[no]).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Flag for coincidence with MXS operation assuming no afterglow, coincidence with antico events, and for recoil and electrical crosstalk, using the time parameters in CALDB. The output file is identical to the input file, but with the STATUS column set according to the flagging results.
sxsflagpix infile=sxs_in.fits inantfile=antico.fits outfile=sxs_out.fits inmxsgti=mxs.gti mxsdt=0

2. Flag for coincidence with MXS operation assuming no afterglow, coincidence with antico events, and for recoil and electrical crosstalk, using the time parameters in CALDB. Also flag events falling within times defined by the intervals in the file hkflag.gti.
sxsflagpix infile=sxs_in.fits inantfile=antico.fits outfile=sxs_out.fits gtifile=hkflag.gti inmxsgti=mxs.gti mxsdt=0

3. Flag, only, for coincidence with MXS operation with afterglow given by CALDB.
sxsflagpix infile=sxs_in.fits inantfile=antico.fits outfile=sxs_out.fits calcprox=no calcant=no calcctrec=no calcctel=no mxfgti=mxs.gti

---

NAME sxsgain -- Calculate the time-dependent energy correction for SXS events from comparison with known calibration lines

USAGE sxsgain infile outfile

DESCRIPTION
'sxsgain' calculates time-dependent SXS energy gain corrections by comparing the theoretical and observed energies of a calibration line or line complex. For each run of the task, only one line may be specified to calculate the gain correction (see parameter 'linetocorrect'). The source of calibration X-rays may be specified as either the SXS calibration pixel ('calmethod=Cal-pix'), the Modulated X-ray Source ('calmethod=MXS'), or the Filter Wheel Fe55 source ('calmethod=Fe55'). In the first case only the calibration pixel (pixel 12) correction is computed, otherwise distinct corrections are computed for each pixel. The output may subsequently be utilized by the 'sxspha2pi' task to assign photon energy and PI values. This task requires that the TIME, PHA, and STATUS columns in the input file are populated.

'sxsgain' accumulates spectra from events that are consecutive in time, with energy centered on the calibration feature and compares each spectrum from the events with a theoretical model of the calibration feature profile. The calibration feature used by 'sxsgain' is specified by the parameter 'linetocorrect' as a string, and the names and energies of the features are specified in a calibration file (parameter 'linefitfile'). The calibration feature may be composed of many atomic or nuclear line components that are listed in the calibration file. Each calibration line is assumed to be Lorentzian.

The energy range for the spectra constructed from the event file as well as from the theoretical profile, may be specified in two different ways. (1) The default energy range for the spectra is the smallest and largest energies of the line components, and expanded with the 'extraspread' parameter, i.e. [E_min - extraspread : E_max + extraspread]. It is recommended to set 'extraspread' larger than the sum of the natural width of the calibration feature, the value of the 'broadening' parameter, and the magnitude of the expected energy shift. (2) Alternatively, the energy range may be specified by setting the 'startenergy' and 'stopenergy' parameters. If these parameters are non-negative 'sxsgain' uses their values to accumulate the spectra, instead of the range derived using the 'extraspread' setting. The profile may be convolved with a Gaussian having the FWHM given by the 'broadening' parameter.

'sxsgain' proceeds in two steps for each spectrum. In the first step, the PHA shift of the spectrum with respect to the theoretical profile is derived, using both fitting and averaging methods. In the second step, these shifts are used to compute pixel temperatures that may subsequently be used by 'sxspha2pi' to assign PI.

Step 1: In order to fit PHA spectra, the spectrum energy grid is converted to a PHA mesh by applying a reverse lookup using the appropriate set of gain coefficients. Gain coefficients are contained in the gain CALDB file (see parameter 'gainfile') for each pixel, resolution grade -- (H)igh, (M)id, and L(ow), and pixel temperature. For a given spectrum in a pixel, the gain corresponding to that pixel, grade given by the 'gaincoeff' parameter, and temperature given by the 'tempidx' parameter are used to define this mesh and to evaluate the theoretical profile on the PHA mesh.

The task proceeds by first constructing spectra within the prescribed PHA range for groups of events consecutive in time and in the same pixel. Only high-resolution primary events, or alternatively high- and mid-resolution primaries (if 'usemp=yes'), are considered. Events may be excluded from the fitting by independently checking the STATUS column for antico, electrical crosstalk, and recoil crosstalk flags if, respectively, 'ckant', 'ckctel', or 'ckctrec' parameters are set to yes. If 'ckrisetime=yes', events with risetime>127 are likewise excluded. If a GTI file (see below GTI format) is specified by the 'gtifile' parameter, events outside of the GTI intervals are excluded as well. If 'calmethod' is set to MXS this GTI should only include times when the MXS mode corresponding to 'linetocorrect' (direct mode for 'linetocorrect=CuKa, CuKb, CrKa, or CrKb'; indirect mode for 'linetocorrect=AlKa, AlKb, MgKa, or MgKb') are producing MXS events. Spectra are not accumulated across GTI intervals unless 'spangti=yes'.

The number of events in each spectrum is defined by the 'numevent' and 'minevent' parameters. The task accumulates spectra with a number of events between 'minevent' and 'numevent'. However, if a spectrum has fewer than 'minevent' points, then it is combined with the previous spectrum if possible. Therefore all spectra have a size between 'minevent' and ('numevent+minevent-1'). To avoid having spectra accumulated over large gaps in time, the group of points in the spectrum is truncated when the time interval between consecutive events is greater than the 'gapdt' parameter. Adjacent spectra in time may share a percentage of their points based on the 'grpoverlap' parameter that may vary between 0 and 100. If 'grpoverlap' is set to 0, the consecutive spectra share no points in common; if set to 100 they share all points in common but one.

For each accumulated spectrum, 'sxsgain' fits the theoretical profile to the data and also derives binned and unbinned averages. A least-squares method is used in the fitting. The fitted parameters are energy shift, scaling factor, background (unless the 'background' parameter is set to NONE), and convolution width if 'fitwidth=yes'. The background is fit with a constant value if 'background' is set to CONSTANT, and a power-law if set to SLOPE. The unbinned average PHA is the sum of the PHA in the spectrum divided by the number of events in the spectrum. The binned average PHA is the weighted average derived by summing, over bins in the spectrum, the product of the PHA and number of events per bin, and then dividing by the total number of events in the spectrum. The fitted gain correction is computed from the fitted shift with respect to the theoretical line profile. The binned average gain correction is computed from the difference between the profile and spectrum averages.

Step 2: Once the fitting is completed for a spectrum in a pixel, the fitted and average pixel temperatures are determined, and written to the output gain file. The gain coefficients contained in the gain CALDB file (parameter 'gainfile') for this pixel and the grade given by the 'gaincoeff' parameter, are used to build a temperature vs. energy table for the average PHA derived from the binned average or from the fitting. The corresponding pixel temperatures at the average energy of the input calibration feature are then obtained using an n-point interpolation method, where n is the number of temperatures in the gainfile to be used (see 'ntemp' parameter).

The default values for the parameters used in the fitting method ('minwidth0', 'maxitcycle', 'r2tol', 'searchstepshift', 'maxdshift', 'bisectolshift', 'searchstepwidth', 'maxdwidth', 'bisectolwidth', and 'minwidth') should not need to change since already optimized.

The output file has two extensions. One extension, GRID_PROFILE, contains the energies and amplitudes of the theoretical profile used in the fitting procedure, including any convolution from the 'broadening' parameter. The other extension, DRIFT_ENERGY, reports the

fitting results for each spectra in the following columns: TIME (midpoint of the time interval over which the spectrum is collected), PIXEL, COR_FIT (energy correction factor from spectrum fit), COR_AVE (energy correction factor from spectrum average), CHISQ (reduced chi-squared of the fit), AVGUNBIN (average energy of events in spectrum prior to binning), AVGBIN (weighted spectrum average energy), AVGFIT (average energy from fit), SHIFT (fitted energy shift), SCALE (fitted vertical scaling factor), BGRND (fitted background), WIDTH (if 'fitwidth=no', same as broadening parameter; if 'fitwidth=yes', fitted width), TELAPSE (difference between times of first and last event in spectrum), EXPOSURE (calculated using the GTI), NEVENT (total number of events collected for this spectrum), BINMESH (array containing the count spectrum energy bins), SPECTRUM (array containing the observed binned count spectrum), FITPROF (array containing theoretical profile with fitted parameters applied), TEMP_FIT (temperature derived from fitted PHA; AVGFIT column), TEMP_AVE (temperature derived from average PHA; AVGBIN column). If the 'calcerr' parameter is set to 'yes', one-sigma errors for the SHIFT and WIDTH are calculated. The errors are calculated with chi-squared and maximum-likelihood methods and output in the columns SIGSHCHI2, SIGWDCHI2, SIGWDLIKE and SIGWDLIKE respectively. If 'writeerrfunc' parameter is set, the chi-squared and likelihood calculated valued are output in the arrays SHCHI2, SHLIKE, WDCHI2 and WDLIKE. The numbers of values output in these arrays are specified in the 'nerrshift' and 'nerrwidth' parameters, respectively.

Two formats are supported for the GTI file. The first format contains up to 37 extensions: 1 general GTI and 36 pixel-dependent GTI. The DETNAM keyword for the general GTI must be PIXEL and the DETNAM keyword values for the pixel-dependent GTIs are PIXnn where nn is the pixel number, e.g. PIX04 for pixel 4. In this format, all extensions must contain columns START and STOP which define the GTI. The second format may have up to two extensions: 1 general GTI with START and STOP columns and 1 pixel-dependent extension with columns START, STOP, and PIXEL. In the second format the DETNAM keyword is set to PIXEL for both extensions. if the input GTI has a single GTI extension with no DETNAM keyword defined, it is treated as a general GTI. If the events need to be filtered using multiple general or pixel-dependent GTI files, they must first be merged into a single GTI file using the 'sxspixgti'.

PARAMETERS

infile [filename]
Input event file name. The file must have the TIME column filled.

outfile [filename]
Output gain correction file name.

(gainfile = CALDB) [filename]
Input file containing SXS energy scale (gain) coefficients as a function of pixel and resolution grade, and pixel temperature. If the parameter is set to CALDB, the file is read from the calibration database.

(tempidx = 2) [integer]
Input temperature index for selecting gain to convert energy to PHA profiles during fitting.

(gaincoeff = H) [string]
Type of gain coefficients to use, H for high-resolution, M for mid-resolution, L for low-resolution.

(linefitfile = CALDB) [filename]
Input CALDB file containing parameters of the Lorentzian components used to construct the theoretical line profile. If the parameter is set to CALDB, the file is read from the calibration database.

(linetocorrect = Mnka) [string]
Calibration line to use for the energy correction. Features in the CALDB file relevant to the SXS are the following: Mnka and Mnkb (cal-pixel and Fe55 filter wheel source); Cuka, Cukb, Crka, and Crkb (MXS direct mode), Alka, Alkb, Mgka, and Mgkb (MXS indirect mode).

(itypecol = ITYPE) [string]
Column name that contains the type event (or grade).

(ntemp = 3) [integer]
Number of temperatures from gain file to use in interpolation. This must not exceed the number of temperatures in gainfile.

(calmethod = Cal-pix) [string]
If calmethod is set to Cal-pix, the energy correction is calculated for the calibration pixel (pixel 12) only. If calmethod is set to MXS the energy correction is calculated for all pixels, but only events obtained during MXS operation, as should be specified by the 'gtifile' parameter, are used. If calmethod is set to Fe55, all pixels and all good times are used.

(numevent = 250) [integer]

Number of events collected for each spectrum used to calculate a single gain correction.

(minevent = 150) [integer]
Minimum number of events required for a spectrum. If the length of a group is less than minevent, those points are included with the previous group for processing, if possible.

(gtiile = NONE) [filename]
Input GTI file with time intervals to consider. Set to NONE to use entire input file.

(gapdt = -1) [real]
The upper limit to the time interval between two consecutive events in the same spectrum used in fitting. Two consecutive events separated in time by more than this amount are assigned to different groups. If gapdt=-1, no limit is imposed.

(grpoverlap = 0) [real]
The percentage overlap between adjacent groups. For grpoverlap=100 adjacent groups are shifted by one event, for grpoverlap=0 adjacent groups are independent and share no events.

(startenergy = -1.) [real]
Beginning of energy range in eV over which the spectra are collected. If startenergy is negative, the first energy is automatically determined by the smallest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(stopenergy = -1.) [real]
End of energy range in eV over which the spectra are collected. If stopenergy is negative, the final energy is automatically determined by the largest energy in linefitfile for the selected calibration feature adjusted by the extraspread parameter.

(extraspread = 10.) [real]
Energy in eV by which the energy range is extended on either side beyond the smallest and largest energy in the linefitfile for the selected calibration feature. This parameter may be overridden by the startenergy and stopenergy parameters.

(pxphaoffset = 0.0) [float]
The PHA bins used to construct the fitting spectra are shifted by this amount. If the PHA value in the gain file represents the left boundary of the PHA bin, then pxphaoffset should be 0; if instead it represents the center of the bin, pxphaoffset should be 0.5.

(broadening = 1.0) [real]
FWHM of the Gaussian in eV used to initially broaden the theoretical line profile. If fitwidth is set to no, the profile width is fixed at this value.

(gridprofile = no) [boolean]
If gridprofile is set to yes, only output the theoretical profile including any convolution due to the broadening parameter; no fitting is conducted (yes/[no]).

(fitwidth = yes) [boolean]
If fitwidth is set to yes, then fit the width of each spectra in addition to the energy shift ([yes]/no).

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE).

(spangti = no) [boolean]
If spangti is set to yes, events in different intervals in gtifile may be collected in the same spectrum to be fit. If spangti is set to no, then groups of events used to construct the spectra must be from the same GTI. This parameter is ignored if gtifile is set to NONE (yes/[no]).

(usemp = no) [boolean]
If usemp is set to no, only high-resolution primaries are included in the fitted spectra. If usemp = yes, mid-resolution primaries are also included (yes/[no]).

(ckrisetime = yes) [boolean]
If ckrisetime is set to yes, events with risetime>127 are not used in fitting ([yes]/no).

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no]).

(writeerrfunc = no) [boolean]
Output the array of chi-squared and likelihood calculated for the SHIFT and WIDTH (yes/[no]).

(ckant = no) [boolean]
If ckant is set to yes, events identified as coincident with antico events according to the STATUS column are not used in fitting (yes/[no]).

(ckctrec = no) [boolean]
If ckctrec is set to yes, events identified as recoil crosstalk according to the STATUS column are not used in fitting (yes/[no]).

(ckctel = no) [boolean]
If ckctel is set to yes, events identified as electrical crosstalk according to the STATUS column are not used in fitting (yes/[no]).

(extrap = no) [boolean]
Allow extrapolation when calculating temperature (yes/[no]).

(avgwinrad = 30) [real]
Radius of interval (in units of binwidth) used only to update the initial shift estimate prior to fitting. This is not used in calculating the average results.

(minwidth0 = 1.0) [real]
Smallest width, in units of binwidth, allowed as the initial value in width fitting. This parameter provides a lower limit to the initial estimate of the width as computed by the fitting algorithm. The value must be greater than zero.

(maxitcylce = 5) [integer]
Maximum number of fitting iterations.

(r2tol = 0.001) [real]
Convergence criterion on R-squared for least-squared fitting. Once R-squared changes by less than this amount between fitting iterations, the procedure is finished. This parameter should not normally need to be changed from the default value.

(searchstepshift = 2.) [real]
Step size, in units of binwidth, used when searching for best-fit energy shift in either direction from the initial shift estimate based on the spectrum average. The final shift is obtained using the bisection method (see bisectolshift).

(maxdshift = 5.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of energy shift. If no solutions are found within this deviation at smaller or larger shifts from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolshift = .001) [real]
When the bisection method determines the energy shift to within this amount in units of binwidth, the fitting procedure is completed.

(searchstepwidth = 5.) [real]
Step size, in units of binwidth, used when searching for best-fit convolution width in either direction from the initial width estimate based on the difference between the profile and spectrum statistical variances. The final width is obtained using the bisection method (see bisectolwidth).

(maxdwidth = 10.) [real]
Largest allowed deviation, in units of binwidth, from initial estimate of convolution width. If no solutions are found within this deviation at smaller or larger widths from the initial estimate, then the fitting procedure fails for the spectrum.

(bisectolwidth = .001) [real]
When the bisection method determines the convolution width to within this amount in units of binwidth, the fitting is procedure is completed.

(minwidth = .5) [real]
Since the least-squares fitting functional is undefined when the width is zero, one must define a minimum allowed fitted width. If the fitting routine attempts to fit a width smaller than this value (in units of binwidth), the fitting procedure fails for the spectrum.

(nerrshift = 100) [int]
Number of shift values in uncertainty calculations.

(nerrwidth = 100) [int]
   Number of width values in uncertainty calculations.

(shifterrfac = 3.0) [real]
   Factor for determining domain of shift uncertainty arrays.

(widtherrfac = 4.0) [real]
   Factor for determining domain of width uncertainty arrays.

(buffer = -1) [integer]
   Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate the gain based on fitting cal-pixel events with a line profile based on the theoretical Mn K-alpha profile convolved with a Gaussian with width determined by fitting.
sxsgain infile=event_in.fits outfile=gain_out.fits

2. Calculate the gain based on fitting cal-pixel events with a line profile based on the theoretical Mn K-alpha profile convolved with a Gaussian of fixed 5 eV FWHM.
sxsgain infile=event_in.fits outfile=gain_out.fits broadening=5.0 fitwidth=no

3. Calculate the gain based on fitting MXS events with a line profile based on the theoretical Cu K-alpha profile using a GTI file indicating when the direct MXS were on, mxson13.gti.
sxsgain infile=event_in.fits outfile=gain_out.fits linetocorrect=Cuka calmethod=MXS gtifile=mxson13.gti

---
NAME sxsmkrmf - Create an SXS RMF file and/or an RSP file for selected SXS pixels and grades

USAGE sxsmkrmf infile outfile resolist regmode region

DESCRIPTION
'sxsmkrmf' is a script that calculates an SXS redistribution matrix file (RMF) for selected grades and pixels properly weighted. The file of weights is calculated with an input event file for the pixel / grade combination. The rmf is calculated by running 'sxsrmf' with the file of weights appropriate for the pixel/grade combination. If an arf file is provided 'sxsmkrmf' output a total response (RSP) file.

The inputs to 'sxsmkrmf' are: (1) a cleaned SXS event file from which spectra are to be extracted; (2) a list of resolution grades (see 'resolist' parameter); and (3) a list of pixels. The selected grades and pixels should match the lists used to extract the spectra and to create the arf.

The pixels selection may be determined in two manners. (1) Input a DS9-format region file ('region') either in DET or SKY coordinates (see 'regmode' parameter). The task 'coordpnt' converts the region into a pixel list using the CALDB teldef file (see 'teldeffile' parameter). By default, the conversion from SKY coordinates uses the pointing recorded in the RA_NOM, DEC_NOM, and PA_NOM keywords of the input event file. However, these may be overridden using the 'rapoint', 'decpoint', and 'roll' parameters. The 'pixeltest' parameter determines the criteria for including pixels based on the region. If 'region=ALLPIX', all pixels are included. (2) If 'region=NONE', the pixel selection is done by entering pixel number in the 'pixelist' parameter.

The selection of grades and pixels determined by the 'sxsmkrmf' parameters is used to construct a file containing the weights for each combination of grade and pixel (sxsfrac.fits) that is utilized by the 'sxsrmf' task to create the RMF file. 'sxsrmf' calculates the RMF for each pixel and grade by using the line spread function (LSF) parameters stored in a CALDB file. The LSF is assumed to consist of (1) a Gaussian core with a pixel-dependent FWHM, (2) a low-energy exponential tail due to energy loss at the surface of the absorbers, (3) an extended low energy electron loss continuum, and (4) discrete escape peaks from M-shell fluorescence of Hg or Te in the absorber. The parameters of the first component are contained in LINESIGMA extension of the CALDB file given by the 'rmfsigma' parameter, and of the latter three components in the LINETAU extension of CALDB file given by the 'rmftau' parameter.

Four types of RMF files may be created. The small (s/S) accounts only for the Gaussian core. The medium (m/M) includes also the exponential tail. The large (l/L) includes the escape peaks and the x-large (x/X) adds the continuum down to the minimum energy given by the 'emincont' parameter. Note that the x-large may exceed the 2.1 GB default limit for the FITS file structure used in response files. Thus the successful creation and application (e.g., in XSPEC) of x-large RMF files cannot be assured, and this option is not recommended for normal usage.

The script writes out an output RMF containing the response matrix; and an EBOUNDS extension which lists the energy boundaries of each PI channel.

The energy bin size(s) of the RMF may be set with the 'dein' parameter, with 'eminin' and 'nchanin' determining the energy range for those bins (see below for more detail). The input energy grid may have multiple values for the number of energies and grid spacing to support a non-uniform grid; however, the single-grid default parameter values of 'eminin=0', 'dein=0.5', and 'nchanin=32768' are recommended. Furthermore, the 'EBOUNDS' energy grid must match the PI grid of the spectrum file. If the parameter 'useingrd=yes', the output grid is set equal to the input grid. This is the recommended binning. However, the 'EBOUNDS' energy grid can be independently specified by setting the parameter values of 'deout', 'eminout', and 'nchanout'. If 'useingrd=yes', the output grid parameters ignored.

If the parameter 'outrsp=yes', an RSP file may be created to combine the RMF, calculated by 'sxsrmf', and the ARF input to sxsmkrmf (see parameter 'arfinfile') . The RSP output filename may be entered in the parameter 'outrspfile'.

PARAMETERS
infile [filename]
Input event file used to calculate the grade and pixel weighting factors. This should be the cleaned event file used to extract the spectrum.

outfile [filename]
Name of output RMF file.

resolist [string]
List of grades (ITYPE). ITYPE=0 for grade Hp, ITYPE=1 for Mp, ITYPE=2 for Ms, ITYPE=3 for Lp, ITYPE=4 for Ls. Multiple numerical values may be entered separated by a comma ,e.g. resolist=1,2,3. If 'resolist' is ALL, all grades are considered.

regmode [string]
Coordinate system for the region file. This is either DET or SKY.

region [string]
Region file name. If region is set to ALLPIX, all pixels are used and regmode is ignored.

(pixlist = 0-35) [string]
List of pixels, in the form of commas separated ranges, e.g. e.g. '0-3,7,13-25,30'. The content of this parameter only used only if region = NONE.

(pixeltest = PARTIAL) [string]
Pixel inclusion option. 'pixeltest=PARTIAL' (the default): if any part of the pixel is within the region, the all pixel is included in the region; 'pixeltest=CENTER': the pixel is included in the region only if the pixel center is within the region; 'pixeltest=TOTAL': the pixel is included in the region only if the entire pixel is within the region.

(rapoint = -999.0) [real]
Pointing RA [deg]. If rapoint is set to -999, the RA is read from the RA_NOM keyword on the event file.

(decpoint = -999.0) [real]
Pointing DEC [deg]. If decpoint is set to -999, the DEC is read from the DEC_NOM keyword on the event file.

(roll = -999.0) [real]
Roll angle [deg]. If roll is set to -999, the roll is read from the PA_NOM keyword on the event file.

(teldeffile = CALDB) [filename]
Input teldef filename.

(outrsp = no) [boolean]
If yes, an output RSP file is created by combining the rmf with the input arf file specified by the 'arfinfile' parameter (yes/[no]).

(outrspfile) [filename]
Name of output response (RSP) file. This parameter is ignored if outrsp=no.

(arfinfile = NONE) [filename]
Name of input ARF file. This parameter is ignored if outrsp=no.

(time = 2014-01-01T00:00:00) [string]
Start time for validity of RMF file in YYYY-MM-DD format. The task selects the CALDB file corresponding to this time.

(whichrmf = L) [string]
Type of RMF to construct -- S(mall), M(edium), L(arge), or X(tra-large). The 'X' option requires is not recommended for general use.

(rmfsigma = CALDB) [filename]
Input file containing the parameters for the Gaussian core. If set to CALDB, the file is read from the calibration database.

(rmftau = CALDB) [filename]
Input file containing the parameters for the exponential tail, electron continuum, and escape peaks. If set to CALDB, the file is read from the calibration database.

(eminin = 0.0) [real]
Minimum energy of the MATRIX grid in eV.

(dein = 0.5) [string]
Spacing of energy bins (in eV) in each interval of the MATRIX grid. A single value (e.g., 0.5 as the default) applies a constant grid spacing for the entire matrix. Multiple values separated by commas (e.g., '0.5,2.0') create a variable grid spacing starting at eminin. The number of channels in each energy interval must be specified with nchanin. The values may be floating point.

(nchanin = 32768) [string]
Number of channels in each interval of the MATRIX grid. Combined with the default values of the previous two parameters, the default means that the output MATRIX has 32768 channels with 0.5 eV size up to 16384 eV. If, instead, dein='0.5,2.0' and nchanin='26000,6500' the output MATRIX would have 26000 bins (with 0.5 eV size) up to 13000 eV, and 6500 bins (with 2 eV size) above 13000 eV (hence up to 26000 eV). dein and nchanin must have the same number of intervals. The values of nchanin must be integers.

(useingrd = yes) [boolean]
If useingrd is set to yes the output grid is set equal to the input grid, and the output grid parameters are ignored. In this case nchanin and dein may only have single values. If useingrd is set to no the output grid is determined by eminout, deout, and nchanout ([yes]/no).

(eminout = 0.0) [real]
Minimum energy of EBOUNDS in eV. The default value is 0.0. This parameter is ignored if useingrd is set to yes.

(deout = 0.5) [real]
Spacing of the EBOUNDS energy grid in eV. This must be consistent with the assumed energy grid of the PI file (i.e., 0.5 eV for the SXS). This parameter is ignored if useingrid is set to yes.

(nchanout = 32768) [integer]
Number of channels in EBOUNDS. This must be consistent with the number of PI channels in the PHA file. The default value for SXS is 32768. This parameter is ignored if useingrd is set to yes.

(rmfthresh = 1.0e-9) [real]
Low threshold for the RMF construction. For a given X-ray photon energy, the normalized probability (i.e., a value less than 1) of producing a certain PI value is calculated for each channel and written in the MATRIX extension. If the probability is less than this threshold, 0 (zero) is written for that element. Using the default value (1.0e-9) is secure enough for most cases.

(emincont = 10.0) [real]
Lower energy limit [eV] of electron continuum.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Create a large-sized RMF file, sxs_allpix_large.rmf, for high-resolution events using all pixels, with weights derived from the cleaned SXS event file, sxs_cl.evt.
sxsmkrmf infile=sxs_cl.evt outfile=sxs_allpix_large.rmf resolist=0 region=ALLPIX

2. Create a small-sized (line-spread function Gaussian core only) RMF file, sxs_region_small.rmf, for high-resolution and mid-resolution primary events, using pixels with any part in the sky region sxs_sky.reg. Also create an rsp file sxs_region_small.rsp using the arf file sxs_region_small.arf.
sxsmkrmf infile=sxs.evt outfile=sxs_region_small.rmf resolist=0,1 region=sxs_sky.reg outrsp=yes outrspfile=sxs_region_small.rsp \
arfinfile=sxs_region_small.arf whichrmf=s

NAME sxsnxbgen -- Create a Non-X-ray Background (NXB) spectrum for SXS

USAGE sxsnxbgen infile ehkfile regfile innxbfile innxbehk innxbhk outpifile

DESCRIPTION
'sxsnxbgen' generates a PI spectrum of the non-X-ray background (NXB) for an SXS observation. The NXB spectrum is calculated from a calibration file containing night Earth data, when X-rays are blocked from the detector's view, accumulated over a period of time. This spectrum is used in spectral fitting (e.g. with XSPEC) to subtract the effects of the particle background from the science data.

'sxsnxbgen' uses the following inputs: (a) the science SXS event file for which the NXB spectrum is calculated ('sxsnxbgen' reads and uses information in the header keywords related to the time of the observation, and the pointing, as well as the GTI); (b) the extended housekeeping (EHK) file ('ehkfile') containing information on the spacecraft orbit during the observation; (c) The NXB event file (parameter 'innxbfile'); the NXB EHK file ('innxbehk'); (d) a DS9-format region file ('regfile') describing the spectral extraction SXS source region. The region may be specified in RAW, DET, FOC, or SKY coordinates (see 'regmode' parameter). For the SXS it is possible to select the region using a list of SXS pixels (see 'pixels' parameters); in this case the user must set 'regfile=NONE'.

'sxsnxbgen' first filters the NXB event data on the region file. This selection is done using DET coordinates and the task internally transforms the coordinates if the region file is not provided in DET. The task then creates a GTI to apply to the NXB event and NXB EHK data using the 'timefirst' and 'timelast' parameters. These extend the NXB GTI beyond that of the science data to ensure sufficient statistics in the output NXB spectrum. A baseline default value, based on experience with previous missions, is to have a window of 300 days centered on the observation. 'sxsnxbgen' then screens the NXB events using the same criteria used for selection of events in the science data (see 'expr' parameter). Finally the NXB spectrum is produced from NXB events within this GTI that are selected and weighted based on the geomagnetic Cut-Off Rigidity (COR), an estimate of the shielding provided by the Earth's magnetic field against impinging charged particles. NXB spectra are extracted based on the distribution of COR that are present in the science event data and weighted by the ratio of the science exposure time in that bin to the total science exposure time. The NXB output spectrum is then the sum of these weighted spectra.

The EHK file contains several Cut-Off Rigidity values. Empirically derived values are stored in the columns COR, COR2, or COR3; and a calculated value in the CORTIME column. The choice of COR to use is specified by the 'sortcol' parameter. COR3 is the recommended table to use for SXS. The 'sortbin' parameter specifies the COR value bin boundaries that are used in the NXB selection.

'sxsnxbgen' outputs the weighted NXB PI spectrum ('outpifile'); and optionally, the EHK file corresponding to the science GTI ('outehkfile'); the calibrated, screened, and time-filtered NXB event list ('outnxbfile'); and the NXB EHK file corresponding to that NXB file ('outnxbehk').

PARAMETERS
infile [filename]
Input event file. Header keywords are read from the EVENTS extension, and the GTI extension is used to construct the weighting histogram.

ehkfile [filename]
Input EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

regfile [filename]
Input region file in DS9 format. The region should be in coordinates specified by regmode. If regfile is set to NONE, the tool uses the pixels parameter to specify the extraction region.

innxbfile [filename]
Input NXB event file used to construct the NXB spectrum.

innxbehk [filename]
Input NXB EHK file. This file must contain the column specified in sortcol used to weight the output NXB spectrum.

outpifile [filename]
Output PI spectrum file name.

(outehkfile = NONE) [filename]
Output EHK file. Contains the EHK data for only the times in the infile GTI. If the parameter is set to NONE this file is not created.

(outnxbfile = NONE) [filename]
Output NXB file. Contains the filtered NXB events used in the output spectrum. If the parameter is set to NONE this file is not created.

(outnxbehk = NONE) [filename]
Output NXB EHK file. Contains the EHK data for only the times covered by the filtered NXB events. If the parameter is set to NONE this file is not created.

(regmode = SKY) [string]
Region mode. Specifies the coordinate system used by regfile. Allowed systems are SKY, DET, FOC, and RAW. If regfile is set to NONE, then this parameter is ignored and pixels is used to define the region.

(timefirst = 150) [integer]
Days before the science observation used to extract NXB.

(timelast = 150) [integer]
Days after the science observation used to extract NXB.

(sortcol = COR3) [string]
Column for sorting NXB data. This column must exist in the EHK files, and must be either COR, COR2, COR3, or CORTIME.

(sortbin = 0,4,5,6,7,8,9,10,11,12,13,99) [string]
Bin boundaries for sorting NXB data. Times where sortcol is below the minimum or above the maximum value is excluded. The range should match any COR filtering performed on the science data. List must be comma-separated in increasing numerical order.

(pixels = 1,2,3,12) [string]
Pixels for sorting NXB data. This is a list of SXS pixels, which is used to filter the NXB data instead of a region. The list of pixels can be comma-separated, or a range can be specified using "-" or ":". The following are equivalent: "1,2,3,12", "1-3,12", and "1:3,12". If pixels="-", then all pixels are used. This parameter is ignored unless regfile is set to NONE.

(expr = NONE) [string]
Additional expression to select good events applied to the NXB event list. This should match the screening of the science data. If the parameter is set to NONE no expression is used.

(cleanup = yes) [boolean]
Delete intermediate files ([yes]/no).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
Run sxsnxbgen with a restrictive COR range, to reduce the background for a low-surface-brightness target.
sxsnxbgen infile=in.evt ehkfile=ehk.fits regfile=target_sky.reg innxbfile=nxb.fits \
innxbehk=nxb.ehk outpifile=target_sxsnxb.pi sortbin="6,7,8,9,10,11,12,13,99"

Run sxsnxbgen using a pixel list instead of a region file.
sxsnxbgen infile=target.evt ehkfile=ehk.fits regfile=NONE innxbfile=nxb.fits \
innnxbehk=nxb.ehk outpifile=target_sxsnxb.pi pixels="1:3,12"

NAME sxspha2pi -- Calculate the PI for the SXS event files

USAGE sxspha2pi infile outfile

DESCRIPTION

'sxspha2pi' calculates and populates the PI column in the SXS event file. 'sxspha2pi' first derived an energy column, EPI and EPI2 (energy in eV), and after calculates the PI column(channel). By default the gain correction uses the PHA column that contains the telemetered values and the computed energy value is written in the EPI column. If the secondary events were corrected using the task 'sxsseccor', a new column containing a combination of the telemetered PHA values for primary events and corrected PHA values for the secondary events is present in the file. The name of the column is PHA2, if 'sxsseccor' is run with the default values. To use this column for the gain correction, users may specify the column name using the parameter 'secphacol'. The ouput of the gain calculation using the 'secphacol' is written in EPI2. NOTE: if the 'secphacol' is set to PHA, the EPI and EPI2 contains the same values. 'sxspha2pi' always calculates the PI value using the EPI2 column.
The EPI (or EPI2) is obtained by using the 'driftfile' derived with the 'sxsgain' task and the gain CALDB file. The gain CALDB file contains gain curves at three temperatures The 'driftfile' provides the relation of PHA versus temperature so that the calibration line is at the correct energy. The relation is computed with two methods average and fit and either result may be used in 'sxspha2pi' (see parameter 'method'). If the drift file is constructed using the calibration pixel (pixel 12), there is only one correction, for each method, that is applied to all pixels. A CALDB scale file may adjust the gain per pixel if specified in the parameter 'scalefile' and if 'scaleepi' is set to yes. If the drift file is constructed using the MXS sources or the Fe55 filter the corrections are for each individual pixel in the array.
'sxspha2pi' reads the 'driftfile' and for each event searches the appropriate correction in time, within a time interval defined by the parameter 'gapdt' and pixel if the driftfile is from the MXS of the FE55 filter. After read the energy, 'sxspha2pi' constructs using the gain table from CALDB curves of temperature versus energy using the event PHA value. Then 'sxspha2pi' reads the 'driftfile' and searches for the temperature, within a time interval defined by the parameter 'gapdt' around the time of the event and pixel if the driftfile is from the MXS of the FE55 filter. The temperature vs Energy table is interpolated to get the EPI corresponding to the temperature from the driftfile. The EPI (or EPI2) column is set to null if there is not corresponding entry in the drift file within the time interval of the event and pixel.. 'sxspha2pi' treats all the events as primary when calculating the EPI. The PI value is calculated as PI=floor((EPI2-0.5)/0.5 + 1.0) For SXS good events, the PI ranges between 0 and 32767 in channel where each channel is 0.5 eV. The PI is set to null if the EPI2 is null or the PI < 0. If the PI > 32767, PI is set to 32767. For SXS baseline event the PI ranges -16384 and 16383. If a baseline event is outside this range the PI is set to null.
The UPI column is calculated if 'calcupi' parameter is set to yes. This column is in energy (eV) derived using the gain file for a single temperature not corrected for the drift. The temperature is defined in the parameter 'tempidx' and by default is set to the value written in the TEMPIDX keyword of the 'driftfile'. For diagnostic purpose the PI calculation may be switched off by setting the 'calcpi' parameter to no, and only the UPI column is populated.
PARAMETERS

infile [filename]
   Input SXS event file name.

outfile [filename]
   Name of the output file name.

(calcupi = yes) [boolean]
   Calculate UPI column ([yes]/no). If 'extend' is et to yes the value are written in a column named UPIE.

(calcpi = yes) [boolean]
   Calculate PI column ([yes]/no).

(driftfile = CALDB) [filename]
   Input file containing drift correction file. This is the output of sxsgaindrift.

(gainfile = CALDB) [filename]
   Input file containing SXS energy scale (gain) coefficients, or CALDB.

(scalefile = CALDB) [filename]
   Input file containing SXS gain adjustments for each individual pixels, or CALDB. This adjustment is only valid if the gain has been calculated using the pixel 12. There are two allowable formats for the scalefile. The first format has two columns, PIXEL and HP, and the scale factors are applied to the event grades as specified in the scalegrade parameter. The second format has three columns, PIXEL, HP, M, and L, giving the specific scale factors for each grade. The scalegrade parameter is ignored when the task is given a scalefile with the second format.

(tempidx = -1) [integer]
   Input temperature index to use in the UPI calculation. If -1 use the temperature index in the 'driftfile'.

(pxphaoffset = 0.0) [float]
   Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+pxphaoffset and +0.5+pxphaoffset. So, when pxphaoffset=0.5, the random offset is between 0 and 1.

(secphacol= PHA) [string]
Input column name to use for the gain correction. The default is PHA containing the telemetered values. If the secondary were corrected (see task sxsseccor) a new PHA column is present in the file (default PHA2) and the user may specified this new column name to apply the gain.

(addepicol = EPI2) [string]
Output energy column name for EPI with secondary correction applied. The value is set to EPI2 and should not be changed. If 'extended' is set to yes, 'sxspha2pi' uses the value in the 'addpicol' as root name and add 'E'(e.g. EPI2E).

(method = FIT) [string]
Correction method (FIT or AVERAGE). If FIT, the TEMP_FIT column in the driftfile is used; if AVERAGE the TEMP_AVG column is used.

(scaleepi= no) [boolean]
Scale EPI for each pixel using the values in the CALDB file entered in the parameter scalefile (yes/[no]).

(scalegrade= 0) [string]
Grade to which the scaling factor per pixel is applied. By default the task applies the scaling factor only to grade 0. Other grades maybe be specified by a comma separated list, e.g. "0,1,2". This parameter is only used when the scalefile contains a single column (HP).

(itypecol = ITYPE) [string]
Column containing event grade in infile.

(extended = no) [boolean]
Use extended energy range (yes/[no]). If the parameter is set to yes, the output of the gain calculation is written in the EPIE and EPI2E columns and the channel is written in the column PIE. NOTE: This option should be used in conjunction with parameters 'binwidth', 'offset' and 'tlmax'. It is advised to use the values recommended in the Hitomi user guide or step by step guide

(binwidth = 1.0) [double]
PI bin width for extended energy range [eV].

(offset = 0.5) [double]
Offset for first PI for extended energy range [eV].

(tlmax = 32767) [int]
Maximum PI channel for extended energy range.

(gapdt = -1.) [double]
Time [s] between events to define a gap (or <0). Events must have entries appropriate to its pixel value in driftfile within this interval; otherwise, EPI and PI are set to their null values.

(ntemp = 3) [int]
Number of temperatures from gain file to use in interpolation. The energy at each of these temperatures is calculated from the PHA of the event, and used to derive the event energy by ntemp-point interpolation.

(writetemp = no) [boolean]
Write temperature used for each event to output event file (yes/[no]).

(extrap = no) [boolean]
Allow extrapolation when determining drift temperature (yes/[no]).

(randomize = yes) [boolean]
If randomize = 'yes', decimal randomization is applied to PHA (an integer) before applying the gain to obtain energy (a double).

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Assign PI for events in event_in.fits. Use the gain (energy scale) coefficients in CALDB and the drift file mxs.drift
sxspha2pi event_in.fits event_out.fits driftfile=mxs.drift.

NAME sxsperseus - Adjust the gain for the Perseus observations

USAGE sxsperseus infile outfile driftfile

DESCRIPTION
'sxsperseus' adjusts the energy scale for the Perseus data. These observations were taken at the very early stage of the mission when the SXS detector was not in thermal equilibrium. To correct the energy scale, these data are first corrected as for all other data with the tasks 'sxsgain' and 'sxspha2pi' using the CALDB gain file that do not include the additional pixel-to-pixel derived from the Fe55 observation. As second step the task 'sxsperseus' calculates the proper gain adjustment derived from the Fe55 to the time of the Perseus observation. The resuls of this calculation are written in the EPIPER and EPI2PER columns. The EPI2PER is used to recompute the PI column. 'sxsperseus' run to all the data collected before 00:41 UT March 4th 2016.

PARAMETERS

infile [file]
Input SXS event file.

outfile [file]
Output SXS event file

driftfile [file]
Input calibration pixel gain file. This is the output of the task 'sxsgain'

(dgfile = REFDATA) [file]
Input gain coefficients file. This file is read from the REFDATA area of HEAsoft and should not be changed.

(offsetfile = REFDATA) [file]
Input calibration offset file. This file is read from the REFDATA area of HEAsoft and should not be changed.

(outrange = NULL) [string]
Define how to handle events with TIMEs outside of the driftfile TIME range. If outrange = NULL, then assign EPIPER=EPI2PER+PI=NULL; if outrange=CONST, then use the gain correction from the first or last time in driftfile; if outrange=EXTRAP, then extrapolate the gain correction from driftfile.

(method = FIT) [string]
Specifies the correction column to use from driftfile. If method=FIT, use the COR_FIT column, if method="AVERAGE", use the COR_AVE column.

(extended = no) [bool]
Use extended energy range (yes/[no]).If the parameter is set to yes, the task read from the column EPIE and EPI2E and write the output of the gain calculation in the EPIEPER and EPI2EPER columns. The channel is written in the column PIE. NOTE: This option should be used in conjunction with parameters 'binwidth', 'offset' and 'tlmax'. It is advised to use the values recommended in the Hitomi user guide or step by step guide

(binwidth = 1.0) [real]
PI bin width for extended energy range [eV]

(offset = 0.5) [real]
Offset for first PI for extended energy range [eV]

(tlmax = 32767) [int]
Maximum PI channel for extended energy range

(buffer = -1) [int]
Rows to buffer (-1=auto, 0=none, >0=numrows)

[ccldhm]

NAME sxspipeline - SXS reprocessing tool

USAGE sxspipeline indir outdir steminputs stemoutputs entry_stage attitude orbit obsgti housekeeping timfile

DESCRIPTION

sxspipeline duplicates most of the pipeline (not trend data) for the SXS. It allows the user to run all or part of the pipeline processing and to vary the calibration files and filtering (screening) criteria used. A number of other pipeline processing parameters can also be changed.

SXS Pipeline Stages

The sxspipeline is divided into 3 stages:

Calibration
Data screening
Product creation

Each stage may be run singly or in combination with the preceding stages. This is controlled by the entry_stage and exit_stage parameters.

SXS Stage 1 consists of the following steps:
(optional) Calculate GTIFOUNDALL (gtiinvert on GTILOST)
Run mxsgti
Calculate gaingti; GTI file:
if linetocorrect eq 'MnKa':Merge GTIOBS GTITEL
else: Merge GTIOBS GTITEL GTIMXSFNON##
Run sxsanticopi
For each event file:
Run coordevt
Run sxsflagpix
Run sxssecid
Calculate GHF
Merge pointing and slew event files
Select merged file
if linetocorrect eq 'MnKa': PIXEL==12&&ITYPE<5&&gtifilter("gaingti;")
else: PIXEL!=12&&ITYPE<5&&gtifilter("<gaingti>")
Run ftsort on filtered merged file
Run sxsgain on filtered, sorted merged file
For each event file:
Run sxspha2pi (GHF input)
Run sxsflagpix
Run sxssecid
Run sxsseccor
Run sxspha2pi (GHF input)
Run sxsperseus
By default, linetocorrect is 'MnKa' and the GHF is calculated using the 'Cal-pix' method.
The possible options for linetocorrect are:
For <calmethod> eq 'Cal-pix':
Set linetocorrect eq 'MnKa'
For <calmethod> eq 'MXS' (LED 1 and LED 3):
Set linetocorrect eq 'CrKa'
Set linetocorrect eq 'CrKb'
Set linetocorrect eq 'CuKa'
Set linetocorrect eq 'CuKb'
For <calmethod> eq 'MXS' (LED 2 and LED 4):
Set linetocorrect eq 'MgKa'
Set linetocorrect eq 'MgKb'
Set linetocorrect eq 'AlKa'

The data screening (Stage 2) is identical to that in the production pipeline, when default parameters are used. For details on the default screening applied to the SXS events (respectively), see:
ahfilter - Create the EHK from attitude & orbital data and create the MKF from housekeeping data based on the CALDB mkfconf file

ahgtigen - Create the GTI from the EHK and MKF parameters based on CALDB selection file
ahscreen - Screen the data based on GTI and CALDB selection file
Default GTI used for screening data are:
GTIPOINT
GTITEL
GTIMXSFNOFF13
GTIMXSFNOFF24
GTIEHK
GTIMKF

The product creation (Stage 3) is identical to that in the production pipeline, when default parameters are used. For SXS events extractor is run on SKY coordinates and a lightcurve, spectra and images are created for each cleaned event file. No gif images are created.

INPUT
The input to sxspipeline is specified using (at minimum) the indir parameter. This should be specified as the sxs event_uf level sequence directory, e.g.:

sxspipeline indir=/path/to/100039010/sxs/event_uf

Paths to specific sxs housekeeping and satellite data files can be specified using the attitude, housekeeping, extended_housekeeping, makefilter, orbit and obsgti parameters. The attitude, orbit and sxs housekeeping files are required for stage 1 calibration.

OUTPUT
Filenames, etc.

The number of output files depends on both pipeline processing stage(s) and the options selected. All output files are written to the directory specified by the outdir parameter. The archive directory structure is NOT reproduced (i.e. all output files are in a single directory).

The names of files produced are the same as those found in the HEASARC archive. However the usual "ahXXXXXXXXX" prefix, where "XXXXXXXXX" is the sequence number, can be replaced by a character string set by the stemoutputs parameter. This defaults to the value set by the steminputs parameter.

PARAMETERS
indir [string]
Directory containing the input data. This should be the path to the event_uf (or event_cl for stage_3 processing)
/path/to/805062010/sxs/event_uf

outdir [string]
Output directory used for all output data, as well as the report file.
CAUTION: If clobber is set, and this directory already exists, then this task overwrites files as needed in this directory.

steminputs [string]
Stem for FITS input files, e.g. ah100039010. This string is used as the base filename for finding input files. For example, if steminputs=ah100039010, then to find the attitude file, ahpipeline matches the regular expression /ah100039010sxs_[ps][\d][0-9]{4}_uf\.evt(\..+)?$/ against all files found in the indir directory.

(stemoutputs = DEFAULT) [string]
Base (stem) output name used for creating output files. If set to "DEFAULT", then steminputs is used.

entry_stage = 1 [1|2|3]
Entry stage, 1 or 2.
Stage 1: Re-calibrate unfiltered event files.
Stage 2: Start from existing unfiltered event files.

exit_stage = 2 [1|2|3]
Exit stage, 1 or 2.
Stage 1: Produces calibrated unfiltered event files.
Stage 2: Produces screened event files.

attitude [string]
Attitude file

(extended_housekeeping = ah1001.ehk) [string]
Extended housekeeping file

(makefilter = ah1001.mkf) [string]
Makefilter file

orbit [string]
Orbit file

obsgti [string]
Observation GTI file

housekeeping [string]
SXS Housekeeping file

timfile [string]
Input TIM file

(sxs_mkflabel = PIXELALL1) [string]
Label to use for SXS MKF GTI creation

(sxs_ehklabel = PIXELALL1) [string]
Label to use for SXS EHK GTI creation

(sxs_evtlabel = PIXELALL1) [string]
Label to use for SXS event screening

(regionfile = NONE) [string]
Input region file

(sxs_start = 0.0) [real]
SXS CALDB start time

(ra = -999.99999) [real]
RA of nominal pointing [deg]

(dec = -999.99999) [real]
Dec of nominal pointing [deg]

(roll = 0.0) [real]
Roll of nominal pointing [deg]

(optdetx = -999.99999) [real]
SXS optical detx coordinate

(optdety = -999.99999) [real]
SXS optical dety coordinate

(optfocx = -999.99999) [real]
SXS optical focx coordinate

(optfocy = -999.99999) [real]
SXS optical focy coordinate

(optskyx = -999.99999) [real]
SXS optical skyx coordinate

(optskyy = -999.99999) [real]
SXS optical skyy coordinate

(ra_pnt = -999.99999) [real]

RA of sxs pointing [deg]

(dec_pnt = -999.99999) [real]
DEC of sxs pointing [deg]

(calc_gtilost = no) [boolean]
Calculate SXS lost off GTI (yes/[no])

(screenlost = no) [boolean]
Screen lost events (yes/[no])

(teldeffile = CALDB) [string]
Input teldef file (or CALDB)

(leapsecfile = REFDATA) [string]
Input leap second file (or CALDB, [REFDATA])

(selectfile = CALDB) [filename]
ahscreen: Input file with the selection expressions

(gainantfile = CALDB) [filename]
Input antico gain file (or CALDB)

(pixdeffile = CALDB) [string]
Input SXS electrical pixel map file (or CALDB)

(gainfile = CALDB) [filename]
Input gain file (or CALDB)

(linefitfile = CALDB) [filename]
Input calibration line file (or CALDB)

(delayfile = CALDB) [filename]
Input instrument delay file (or CALDB)

(sxs_pulsefile = CALDB) [filename]
Input file with pulse amplitudes (or CALDB)

(adrgti = REFDATA) [string]
Input ADR GTI file (or [REFDATA])

(acphaoffset = 0.0) [float]
Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+acphaoffset and +0.5+acphaoffset. So, when acphaoffset=0.5, the random offset is between 0 and 1.

(pxphaoffset = 0.0) [float]
Average offset added to PHA values before applying the gain. A random offset is added to each PHA between -0.5+acphaoffset and +0.5+acphaoffset. So, when acphaoffset=0.5, the random offset is between 0 and 1.

(stimecol = S_TIME) [string]
Name of S_TIME column

(tioncol = FWE_TI_LED#_ON) [string]
Input TI columns with LED on (#=1-4)

(tioffcol = FWE_TI_LED#_OFF) [string]
Input TI columns with LED off (#=1-4)

(plslencol = FWE_LED#_PLS_LEN) [string]
Input pulse length columns (#=1-4)

(plsspccol = FWE_LED#_PLS_SPC) [string]

Input pulse spacing columns (#=1-4)

(timeoncol = TIME_LED#_ON) [string]
Output LED-on time columns (#=1-4)

(timeoffcol = TIME_LED#_OFF) [string]
Output LED-off time columns (#=1-4)

(calctime = yes) [boolean]
Perform time assignment ([yes]/no)

(calcgti = yes) [boolean]
Produce GTI files ([yes]/no)

(afterglow = no) [boolean]
Add afterglow to fine GTI STOP times (no/[yes])

(dtdecay = CALDB) [string]
Afterglow time [s] (or CALDB)

(interp = twopoint) [string]
Interpolation method (NEAREST, TWOPOINT)

(margingti = yes) [boolean]
Create GTI between TSTART/TSTOP and first/last input GTI

(tstart = DEFAULT) [string]
Value to use for TSTART in seconds (or take from infile)

(tstop = DEFAULT) [string]
Value to use for TSTOP in seconds(or take from infile)

(dt = 0.) [real]
Time separation between input and output GTI (seconds)

(dattfile = datt.out) [string]
output datt file with drift corrections

(coordevt_startsys = LOWEST) [string]
Starting coordinate system

(stopsys = HIGHEST) [string]
Final coordinate system

(annaber = no) [string]
Apply annual aberration correction (yes, [no], INVERT)

(followsun = no) [boolean]
Recalculate the Sun position for each event (yes, [no])

(orbaber = no) [string]
Apply sat orbital aberration correction (yes, [no], INVERT)

(attinterp = LINEAR) [string]
Sky attitude interpolation method (LINEAR, CONSTANT)

(dattinterp = LINEAR) [string]
Delta attitude interpolation method (LINEAR, CONSTANT)

(attdt = 32.) [real]
Allowed margin for time extrapolation in attfile [s]

(dattdt = 0.5) [real]
Allowed margin for time extrapolation in dattfile [s]

(chkattgap = no) [boolean]
Limit attitude interpolation if gaps present (yes, [no])

(chkdattgap = yes) [boolean]
Limit delta attitude interpolation if gaps present ([yes], no)

(attext = ATTITUDE) [string]
Attitude extension

(attcol = QPARAM) [string]
Attitude column

(attform = QUAT) [string]
Attitude format ([QUAT], EULER)

(orbext = ORBIT) [string]
Orbit extension

(orbcol = VELOCITY) [string]
Orbital velocity column

(orbform = VECTOR) [string]
Orbital velocity format ([VECTOR], COMPONENTS, KEPLERIAN)

(coordevt_randomize = TELDEF) [string]
Randomize coordinates when rebinning ([TELDEF], yes, no)

(randsys = TELDEF) [string]
Starting system for randomization (or TELDEF)

(randscalesys = TELDEF) [string]
System to determine randomization amount (or TELDEF)

(infileext = EVENTS) [string]
Event extension

(timecol = TIME) [string]
Time column

(inclfloatcol = no) [boolean]
Write non-rounded coordinate columns (yes, [no])

(inclfloatskycol = no) [boolean]
Write non-rounded sky coordinate columns (yes, [no])

(floatcolsuffix = _FLOAT) [string]
Suffix for non-rounded coordinate columns

(startwithfloat = no) [boolean]
Start with non-rounded startsys coordinates (yes, [no])

(blankcol = yes) [boolean]
Assign null values to columns not calculated ([yes], no)

(btnull = 255) [integer]
TNULL for byte (B) columns

(itnull = -999) [integer]
TNULL for short (I) columns

(jtnull = -999) [integer]
TNULL for long (J) columns

(ktnull = -999) [integer]
TNULL for long (K) columns

(sbtnull = 255) [integer]
TNULL for signed byte columns

(uitnull = -999) [integer]
TNULL for unsigned short columns

(ujtnull = -999) [integer]
TNULL for unsigned long columns

(antpsp = A) [string]
Antico PSP to use for coincidence (A=PSPA B=PSPB)

(antshift = CALDB) [string]
Time shift [s] to apply to antico events (or CALDB)

(calcant = yes) [boolean]
Flag antico events ([yes]/no)

(antdtpre = CALDB) [string]
Delta time [s] preceding an antico event (or CALDB)

(antdtfol = CALDB) [string]
Delta time [s] following an antico event (or CALDB)

(antswitch = 1) [integer]
If =1 use antdtfol, =0 read delta-time from file

(antphathr = 1) [integer]
PHA threshold for antico events

(antdurthr = 1) [integer]
DURATION threshold for antico events

(calcctrec = no) [boolean]
Flag recoil cross-talk ([yes]/no)

(ctrecdt = CALDB) [string]
Delta time [s] for flagging recoil cross-talk (or CALDB)

(calcprox = yes) [boolean]
Flag electrical cross talk ([yes]/no)

(proxdt = CALDB) [string]
Delta time [s] to define simultaneous events (or CALDB)

(calcctel = yes) [boolean]
Flag electrical cross talk ([yes]/no)

(cteldt = CALDB) [string]
Delta time [s] for flagging electrical cross-talk (or CALDB)

(ctelnear = 1) [integer]
Number of pixels for flagging electrical cross-talk

(calcctel2 = no) [boolean]

Flag electrical cross talk 2 ([yes]/no)

(cteldt2 = CALDB) [string]
Delta time [s] for flagging electrical cross-talk 2 (or CALDB)

(ctelnear2 = 1) [integer]
Number of pixels for flagging electrical cross-talk 2

(pxpthr = 600) [integer]
Events with PI values below this threshold are excluded from flagging checks given by the usepxpithr parameter.

(usepxpithr = ALL) [integer]
A comma-delimited list specifying which flagging types should use the pxpithr parameter for excluding events. Allowed values in the list are ALL, NONE, PROX (proximity), CTEL (electrical cross talk), CTEL2 (2nd electrical cross talk), and CTREC (recoil cross talk). Events that do not belong to the types specified in the list are excluded from flagging regardless of their PI value.

(calcmxs = yes) [boolean]
Flag MXS pixels ([yes]/no)

(mxsdt = CALDB) [string]
Delta time [s] to extend MXS stop time (or CALDB)

(kalow = 5860.) [real]
Lower energy limit of Mn K-alpha for recoil PHA test [eV]

(kahigh = 5930.) [real]
Upper energy limit of Mn K-alpha for recoil PHA test [eV]

(kbeta = 6450.) [real]
Energy of Mn K-beta for recoil PHA test [eV]

(dtflag = no) [boolean]
Add delta-time columns for cross-talk and antico (yes/[no])

(dtprimary = CALDB) [string]
Time interval [ms] for primary (or CALDB)

(dtlowmid = CALDB) [string]
Upper time range [ms] for low secondary (or CALDB)

(dtmidhigh = CALDB) [string]
Upper time range [ms] for mid secondary (or CALDB)

(tol = 2.) [real]
Tolerance of time intervals [ns]

(regrade = no) [boolean]
Recalculate grade assignment (yes/[no])

(sxs_resetflags = yes) [boolean]
Reset all sxsflagpix STATUS flags ([yes]/no)

(phaout = PHA2) [filename]
Name of output PHA column

(gaincoeff = H) [string]
Type of gain coefficients to use ([H]/M/L)

(linetocorrect = Mnka) [string]
Line to fit (HDU name in linefitfile)

(numevent = 250) [integer]

Maximum number of events in a single spectrum

(minevent = 150) [integer]
Minimum number of events in a single spectrum

(grpoverlap = 0.) [real]
Percentage of overlap between adjacent groups

(startenergy = -1.) [real]
Start energy [eV] of bin mesh (-1 = automatic)

(stopenergy = -1.) [real]
Stop energy [eV] of bin mesh (-1 = automatic)

(extraspread = 100.) [real]
Extend bin mesh energy range [eV]

(broadening = 1.0) [real]
FWHM Gaussian broadening of calibration line profile [eV]

(gridprofile = no) [boolean]
Calculate only the grid profile (yes/[no])

(fitwidth = yes) [boolean]
Fit spectrum width (yes/[no])

(background = CONST) [string]
Fitted background type (NONE, CONST, SLOPE)

(spangti = no) [boolean]
Ignore GTI boundaries when binning spectra (yes/[no])

(usemp = no) [boolean]
Include Mp events when fitting (yes/[no])

(calcerr = no) [boolean]
Compute uncertainties on shift and width (yes/[no])

(writeerrfunc = no) [boolean]
Output uncertainty functions (yes/[no])

(avgwinrad = 30) [real]
Radius of interval [binwidth] used to update average

(minwidth0 = 1.0) [real]
Smallest allowed initial value in width fitting [binwidth]

(maxitcycle = 5) [integer]
Maximum number of fitting iterations

(r2tol = .01) [real]
Convergence criterion for R^2

(searchstepshift = 2.) [real]
Step size when fitting shift [binwidth]

(maxdshift = 5.) [real]
Largest allowed deviation from initial guess of shift [binwidth]

(bisectolshift = .1) [real]
Tolerance of shift to stop bisection method [binwidth]

(searchstepwidth = 5.) [real]
Step size when fitting width [binwidth]

(maxdwidth = 10.) [real]
Largest allowed deviation from initial guess of width [binwidth]

(bisectolwidth = .2) [real]
Tolerance of width to stop bisection method [binwidth]

(minwidth = .5) [real]
Smallest width to allow in width fitting [binwidth]

(nerrshift = 100) [integer]
Number of shift values in uncertainty calculations

(nerrwidth = 100) [integer]
Number of width values in uncertainty calculations

(shifterrfac = 3.0) [real]
Factor for determining domain of shift uncertainty arrays

(widtherrfac = 4.0) [real]
Factor for determining domain of width uncertainty arrays

(calcupi = yes) [boolean]
Calculate UPI column ([yes]/no)

(scalefile = CALDB) [filename]
Input EPI scale file for cal-pix (or CALDB)

(secphacol = PHA) [string]
Input PHA column to use for secondary correction

(scaleepi = no) [boolean]
Scale EPI values using scalefile (yes/[no])

(scalegrade = 0) [string]
List of grades to apply scale factors

(calcpi = yes) [boolean]
Calculate PI column ([yes]/no)

(addepicol = EPI2) [string]
Output energy column with secondary correction

(method = FIT) [string]
Correction method (FIT or AVERAGE)

(extended = no) [boolean]
Use extended energy range (yes/[no])

(binwidth = 0.5) [real]
PI bin width for extended energy range [eV]

(offset = 0.5) [real]
Offset for first PI for extended energy range [eV]

(tlmax = 32767) [integer]
Maximum PI channel for extended energy range

(writetemp = no) [boolean]
Output temperature used for each event (yes/[no])

(dgfile = REFDATA) [file]
Input gain coefficients file

(offsetfile = REFDATA) [file]
calibration offset file

(outrange = NULL) [file]
How events are handled outside time range

(itypecol = ITYPE) [string]
ITYPE column

(ckctrec = no) [boolean]
Exclude events with recoil cross-talk (yes/[no])

(ckctel = no) [boolean]
Exclude events with electrical cross-talk (yes/[no])

(ckant = no) [boolean]
Exclude events with antico coincidence (yes/[no])

(ckrisetime = yes) [boolean]
Do not use events with RISE_TIME > 127 ([yes]/no)

(tempidx = 2) [integer]
Input temperature index for selecting gain

(ntemp = 3) [integer]
Number of temperatures from gain file to use in interpolation

(gapdt = -1.) [real]
Time [s] between events to define a gap

(extrap = no) [boolean]
Allow extrapolation when determining drift temperature (yes/[no])

(randomize = yes) [boolean]
Allow randomization (yes, no)

(seed = 0) [integer]
Random number generator seed (0=use system time)

(stemreport =) [string]
File stem for log and temporary files. If the parameter is not set the script automatically set the stem to "sxspipeline_YYYYMMDDTHHMMSS_" and appends log file and temp file names as needed. Intended to be set by ahpipeline.

(numerrs = 0) [string]
Number of errors from sxspipeline (output)

(cleanup = yes) [boolean]
Delete temporary files ([yes]/no)

[ccldhm]

EXAMPLES
1.The following command recalibrates (stage 1) and re-screen (stage 2) all SXS data for sequence 100039010 that currently resides in the directory /data/100039010/sxs/event_uf, and the output is stored in a directory called /data/100039010_reproc/, as well as recalculate the SXS lost off GTI:
sxspipeline indir=/data/100039010/sxs/event_uf outdir=/data/100039010_reproc entry_stage=1 exit_stage=2 steminputs=ah100039010 \
attitude=/data/100039010/auxil/ah100039010/ah100039010.att orbit=/data/100039010/auxil/ah100039010/ah100039010.orb \
obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti housekeeping=/data/100039010/sxs/hk/ah100039010sxs_a0.hk1 \

makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk calc_gtilost=yes

2. The following command re-screens (stage 2 only) SXS data for the same data set as in the previous example:
sxspipeline indir=/data/100039010/sxs/event_uf outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 steminputs=ah100039010 \
obsgti=/data/100039010/auxil/ah100039010/ah100039010_gen.gti makefilter=/data/100039010/auxil/ah100039010/ah100039010.mkf \
exended_housekeeping=/data/100039010/auxil/ah100039010/ah100039010.ehk

3.The following command creates products (stage 3 only) SXS data for a calibrated data set:
sxspipeline indir=/data/100039010/sxs/event_cl outdir=/data/100039010_reproc entry_stage=2 exit_stage=2 steminputs=ah100039010\
regionfile=none

NOTES
None, but see help for individual parameters above.

NAME sxspixgti - Create SXS GTI files

USAGE sxspixgti mkffile outfile outpixfile selectfile

DESCRIPTION
'sxspixgti' creates SXS pixel dependent GTI file by merging GTI that are not pixel dependent with pixel depenedent GTI. The output file includes two extensions: the 'good' pixel GTI (EXTNAME=GTIPIXEL) and a 'bad' pixel GTI (EXTNAME=GTIPIXELOFF). The input pixel dependent GTIs are defined either in the telemetered lost GTI file (parameter gtilost) or by pixel depenedent expressions applicable to the MKF file (paramaters mkffile , selectfile, label). These GTIs are merged with others not pixel dependent GTI (parameter gtifile). Therefore the inputs to 'sxspixgti' are either the MKF file or a GTI lost file or both as well as a GTIs file non pixel dependent. If the MKF file is set to NONE, the GTI lost file must be input and vice-versa.
The input GTI lost file is expected to have an extension named GTILOST, the keywords DETNAM set to PIXEL and the three columns START STOP and PIXEL. 'sxspixgti' uses 'gtiinvert' to create the invert either good or bad GTI. The lost GTI are inverted per pixel selecting the start and stop appropriate to the pixel number from the PIXEL column (a). These 36 GTIs are combined with other GTI and create the GTIPIXEL extension in the output.
The GTI pixel dependent expression are stored in CALDB in the select file. These expressions are used to create (b) good pixel GTI. The (b) GTI are merged with (a) GTI and other GTI to create the GTIPIXEL extension.

'ahgtigen' is used to create and merge the good GTI per pixel. The good GTI are merged running 36 times one per pixel using mkffile, the inverse per pixel of the GTI LOST and gtifile. The label parameter is used in 'ahgtigen' with the CALDB selectfile to calculate GTI from the mkffile. By default label is set to PIXELEXP##, where ## is replaced with a pixel number, e.g. PIXELEXP00, PIXELEXP01, etc. The 36 GTI pixel extensions are combined into a single extension and the column PIXEL is populated with the pixel number appropriate for that GTI interval. The bad pixel GTI are created by inverting the 36 extensions and combining in a sigle extension with the PIXEL column populated with the pixel number appropriate for that GTI interval.

The task creates one output file with two extensions each containg three columns START and STOP and PIXEL. The extension GTIPIXEL contains the 'good' GTI and the extension GTIPIXELOFF the 'bad' pixel GTI. Both GTIPIXEL and GTIPIXELOFF are sorted by START then STOP then PIXEL and 'coordevt' is run to calculate coordinates up to FOC. The extension GTIPIXELOFF is used in the 'ahexpmap'.

By setting the parameter outfile different from NONE (default), the final start and stop per pixel calculated in the process are saved in two files, one for the good GTI and the other with the bad GTI, both containing 36 extensions one per pixel. These extension only contains START and STOP. The extensions for each pixel are named GTIPIXnn, e.g. GTIPIX00, GTIPIX01, etc. for the good pixel gti and GTIPIXOFFnn, e.g. GTIPIXOFF00, GTIPIXOFF01 etc. for the bad pixel gti.

PARAMETERS
mkffile [filename]
Input MKF file

outfile [filename]
Root name of the 36-ext GTI file. To the root name is addeded "_good.gti" or " _bad.gti" for the good and bad GTI respectively

outpixfile [filename]
Output file containing two extensions, GTIPIXEL (good) and GTIPIXELOFF (bad), where the PIXEL column list the pixel number appropriate for that START and STOP

selectfile [string]
Input expression file

(gtilost = NONE) [filename]
Input GTI lost file (or NONE)

(label = PIXELEXP##) [string]
Screening expression label in labelfile (## replaced with pixel number)

(gtiexpr = NONE) [string]
Additional GTI expression (or NONE)

(gtifile = NONE) [filename]
Input GTI file (or NONE)

(mergegti = AND) [string]
Merge mode (OR, [AND])

(upkeyword = yes) [boolean]
Update timing keywords ([yes],no)

(leapsecfile = REFDATA) [string]
Input leap second file (or CALDB, [REFDATA])

(instarts = START) [string]
Input column with GTI start times

(instops = STOP) [string]
Input column with GTI stop times

(time = TIME) [string]
Input column with HK parameter times

(outstart = START) [string]
Output column with GTI start times

(outstop = STOP) [string]
Output column with GTI stop times

(prefr = 0.) [real]
Pre-time interval factor [0,1]

(postfr = 1.) [real]
Post-time interval factor [0,1]

(teldeffile = CALDB)[string]
Name of the Telescope Definition (TelDef), specifying the coordinate systems and transformation properties. If the parameter is set to CALDB, the default, a TelDef file for the telescope and SXS is read from the calibration database.

(randomize = TELDEF) [string]
If this parameter is set to 'no', 'coordevt' assumes that each event occurred at the center of its coordinate pixel. If 'randomize=TELDEF', the default, randomization depends on the keyword RANCOORD in the TELDEF file ('RANCOORD = NONE' disables randomization). If 'randomize=yes', the coordinates from system 'randsys' (see below) onward is calculated assuming a random location within the randsys system pixel. This parameter only controls randomization in transformations previous to the transformation to the SKY system.

(seed = 0) [integer]
Random number generator seed; uses system time for seed=0.

[ccldhm]

EXAMPLES
1. The following command calculates GTI for SXS pixels using the selection file in CALDB and the default label of PIXELEXP##:

sxspixgti mkffile=in.mkf outfile=sxs_pixels.gti outpixfile=sxs_pxexpo.gti selectfile=CALDB

NOTES
None, but see help for individual parameters above.

NAME sxsregext - Extract SXS data products from an event file using a region and selection of grades

USAGE sxsregext infile regmode region outroot outexp ehkfile delta numphi

DESCRIPTION
'sxsregext' is a script that takes an SXS event file and region in RA/DEC or DET coordinates and produces a spectrum, an image, and a lightcurve for a selection of event grades. The spectrum and lightcurve are extracted in detector coordinates. If the input region is in RA/DEC coordinates, the script converts the region to one in DET coordinates that includes any pixel in the region for any of the attitude histogram bins in the exposure map that the script creates.

The inputs to 'sxsregext' are: (1) an SXS event file from which data products are to be extracted; (2) a list of resolution grades (see 'resolist' parameter); (3) a region file in RA/DEC or DET coordinates (see 'regmode' and 'region' parameters); and (4) an extended housekeeping (EHK) file (see 'ehkfile' parameter) that is needed to create the exposure map. If the input region is in RA/DEC coordinates, the script runs the 'ahexpmap' and 'ahmkregion' tasks to create an output exposure map (see 'outexp' parameter), and an output region file in DET coordinates with prefix given by the 'outroot' parameter. The region includes any pixel that falls in the region for any of the attitude histogram bins in the exposure map (see 'ahexpmap' help file). An image, spectrum, and lightcurve filtered on the list of grades, are created (see outroot parameter). If 'regmode=DET' the spectrum and lightcurve are extracted from the input region; if 'regmode=RADEC' the spectrum and lightcurve are extracted using the newly-created output region file (also with prefix given by 'outroot'). The identical detector region, selection of grades, and exposure map should be used to generate the RMF (see 'sxsmkrmf' help file) and ARF (see 'aharfgen' help file). The 'ahexpmap' input parameters 'delta' and 'numphi' that determine the resolution of the attitude histogram are also required (see 'ahexpmap' help file).
PARAMETERS

infile [filename]
Input SXS event file.

regmode [string]
Coordinate system for the region file. This is either DET or RADEC. If this is RADEC the region must be in FK5 (RA and Dec).

region [filename]
Region file name. The format is that of a standard SAO region file.

outroot [string]
Root of output file names. If created, image, spectrum, exposure map, and output detector region are given this prefix.

(extract = yes) [boolean]
If yes, extract image, lightcurve, and spectrum. All extracted products are filtered on grade according to the resolist parameter; the spectrum and lightcurve are also filtered on the input region if regmode=DET and the output region if regmode=RADEC. The parameter must be set to yes if regmode=DET (yes/[no]).

(resolist=0,1) [string]
List of grades (ITYPE). ITYPE=0 for grade Hp, ITYPE=1 for Mp, ITYPE=2 for Ms, ITYPE=3 for Lp, ITYPE=4 for Ls. Multiple numerical values may be entered separated by a comma, e.g. resolist=1,2,3. If 'resolist' is ALL, all grades are considered.

(teldeffile = CALDB) [filename]
Input teldef filename.

outexp [filename]
Name of output exposure map file. This parameter is ignored if regmode=DET.

ehkfile [filename]
Name of input EHK file. This parameter is ignored if regmode=DET.

delta [real]
Size in arcmin of each off-axis annulus grid (and on-axis circle) in the output histogram. This parameter is ignored if regmode=DET.

numphi [integer]
Number of azimuth (phi) bins in the first off-axis annulus at (delta <= theta < 2 * delta). A n-th off-axis annulus has azimuth bins of numphi * n. This parameter is ignored if regmode=DET.

badimgfile [filename]
Name of input bad pixel image file. The file should contain an image in the primary extension with the following flag values: good pixels (0), calibration source region (1), bad pixels and columns (2), and out of the detector or area discrimination (-1, null). For the SXI, this is simply the output of the sxiflagpix routine ('outbadimg'). This parameter is ignored if regmode=DET.

pixgtifile [filename]
Name of input pixel GTI list. The file contains a bin table with columns of START, STOP, DETX, and DETY, indicating the duration and location of each partial bad pixel. This parameter is ignored if regmode=DET.

(instmap = CALDB) [filename]
Name of input instrument map file. If the parameter is set to CALDB, the file is read from the calibration database. This file is required for the task to run. This parameter is ignored if regmode=DET.

(qefile = CALDB) [filename]
Name of input quantum efficiency file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(contamifile = CALDB) [filename]
Name of input contamination file. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(obffile = CALDB) [filename]
Name of input optical blocking file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(fwfile = CALDB) [filename]
Name of input filter wheel file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(gvfile = CALDB) [filename]
Name of input gate valve file for the SXS. If the parameter is set to CALDB, the file is read from the calibration database. If the parameter is set to NONE, the task does not use this calibration information. This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(maskcalsrc = yes) [boolean]
If this parameter is set to 'yes', calibration source regions are regarded as bad pixels and excluded from the output exposure map. This parameter is ignored when 'badimgfile' is set to NONE, or if regmode=DET.

(fwtype = OPEN) [string]
Filter wheel type for the SXS (OPEN, FE55, BE, ND, or POLY) This parameter is ignored when 'outmaptype' is set to EXPOSURE, or if regmode=DET.

(specmode = MONO) [string]
Type of input energy: monochrome energy (MONO) or spectrum (SPEC). This parameter is ignored if regmode=DET.

(specfile) [filename]
Name of input spectrum file. This parameter is ignored if 'specmode' is set to MONO. This parameter is ignored if regmode=DET.

(specform = FITS) [string]
Format of the input spectrum file (FITS or ASCII). This parameter is ignored if 'specmode' is set to MONO. This parameter is ignored if regmode=DET.

(extended =no) [boolen]

Use the column PI if 'extended' is set to no to derive the spectrum. If 'extended' set to yes uses the column PIE. The PI and/or PIE columns must be present and populated in the event file.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows)
[ccldhm]

EXAMPLES
1. Extract an image (sxsdata.img), spectrum (sxsdata.pha), and lightcurve (sxsdata.lc) for grades HP and MP from event file sxs.evt. Calculate and output (sxsdata.reg) the detector region used to extract the spectrum and lightcurve using the SKY region sxs_wcs.reg and the EHK file sxs.ehk, and create the exposure map file sxsexpmap.fits.
sxsregext infile=sxs.evt regmode=SKY region=sxs_wcs.reg outroot=sxsdata \
outexp=sxsexpmap.fits ehkfile=sxs.ehk delta=0.5 numphi=4

2. Extract an image (sxsdata.img), spectrum (sxsdata.pha), and lightcurve (sxsdata.lc) for grade HP only from event file sxs.evt. Calculate and output (sxsdata.reg) the detector region used to extract the spectrum and lightcurve using the SKY region sxs_wcs.reg and the EHK file sxs.ehk, and create the exposure map file sxsexpmap.fits.
sxsregext infile=sxs.evt regmode=SKY region=sxs_wcs.reg outroot=sxsdata resolist=0 \
outexp=sxsexpmap.fits ehkfile=sxs.ehk delta=0.5 numphi=4

3. Extract an image (sxsdata.img), spectrum (sxsdata.pha), and lightcurve (sxsdata.lc) for grades HP and MP from event file sxs.evt using the detector region sxs_det.reg.
sxsregext infile=sxs.evt regmode=DET region=sxs_det.reg outroot=sxsdata outexp=none ehkfile=none delta=0.5 numphi=4

NAME sxsrmf - Create an SXS RMF file for selected SXS pixels and grades with weighting factors

USAGE sxsrmf infile outfile

DESCRIPTION
'sxsrmf' calculates the RMF for each SXS pixel and resolution grade (High, Mid, or Low), by using the line spread function (LSF) parameters stored in two extensions of a CALDB file. Then the task combines these RMFs into one RMF file according to an input file with weighting factors for combinations of pixel and grade. The weighting factor file is created by the 'sxsmkrmf' script, and consist of a fits file with three columns (PIXEL, GRADE, WEIGHT), where WEIGHT is defined as the ratio of counts for the specific PIXEL and GRADE combination of the rmf generation to the total in all selected PIXELS.

'sxsrmf' may be run to create an RMF for any SXS pixel and grade combination. This is archived by setting 'infile=none' and use the parameters 'pixel' and 'resol' to select the pixel and the grade, respectively. If the 'outrsp' parameter is 'yes' and an ARF file is given with the 'arffile' parameter, the task also calculates and outputs an RSP file with filename given by the 'outrspfile' parameter.

The LSF consists of (1) a Gaussian core with a pixel-dependent FWHM, (2) a low-energy exponential tail due to energy loss at the surface of the absorbers, (3) an extended low energy electron loss continuum, and (4) discrete escape peaks from M-shell fluorescence of Hg or Te in the absorber. The latter three components are assumed to be pixel-independent. All these components are stored in CALDB file in the 'linesigma' and 'linetau' tables . Four types of RMF files may be created small, medium, large and x-large. The small (s/S) accounts only for the Gaussian core calculated with a width, defined in the CALDB linesigma table (see 'rmfsigma' parameter). The medium (m/M) includes also the exponential tail for the line defined in the caldb linetau table (see 'rmftau' parameter); the large (l/L) includes the escape peaks. The x-large (x/X) adds the continuum down to the minimum energy given by the 'emincont' parameter. Note that the x-large may exceed the 2.1 GB default limit for the FITS file structure used in response files. Thus the successful creation and application (e.g., in XSPEC) of x-large RMF files cannot be assured, and this option is not recommended for normal usage.

The output RMF contains a MATRIX extension containing the response matrix; and an EBOUNDS extension listing the energy boundaries of each PI channel.
The energy bin size(s) of the RMF may be set with the 'dein' parameter, with 'eminin' and 'nchanin' determining the energy range for those bins (see below for more detail). The input energy grid may have multiple values for the number of energies and grid spacing to support a nonuniform grid; however, the single-grid default parameter values of 'eminin=0', 'dein=0.5', and 'nchanin=32768' are recommended. Furthermore, the 'EBOUNDS' energy grid must match the PI grid of the spectrum file. If the parameter 'useingrd=yes', the output grid is set equal to the input grid. This is the recommended binning. However, the 'EBOUNDS' energy grid can be independently specified by setting the parameter values of 'deout', 'eminout', and 'nchanout'. If 'useingrd=yes', the output grid parameters ignored.

PARAMETERS
infile [filename]

Name of input FITS file containing the weighting factor. If set to NONE the rmf for a single pixel and grade is created corresponding to the settings of the pixel and resol parameters, respectively.

outfile [filename]
Name of output RMF file.

(outrsp = no) [boolean]
If yes, an output RSP file is created by combining the rmf with the input arf file specified by the arffile parameter (yes/[no]).

(arffile = NONE) [filename]
Name of input ARF file. This parameter is ignored if outrsp=no.

(rspfile) [filename]
Name of output response (RSP) file. This parameter is ignored if outrsp=no.

(pixel = 0) [integer]
SXS array pixel (0-35) for which the RMF is calculated. For each run only one pixel maybe be input. This parameter is ignored unless infile is set to NONE.

(resol = H) [string]
SXS event resolution grade -- (H)igh, (M)id, or (L)ow -- for which the RMF is calculated. For each run only one grade maybe be input. This parameter is ignored unless infile is set to NONE.

(time = 2014-01-01T00:00:00) [string]
Start time for validity of RMF file in YYYY-MM-DD format. The task selects the CALDB file corresponding to this time.

(whichrmf = L) [string]
Type of RMF to construct -- S(mall), M(edium), L(arge), or X(tra-large). The 'X' option is not recommended for general use.

(rmfsigma = CALDB) [filename]
Input file containing the parameters for the Gaussian core. If set to CALDB, the file is read from the calibration database.

(rmftau = CALDB) [filename]
Input file containing the parameters for the exponential tail, electron continuum, and escape peaks. If set to CALDB, the file is read from the calibration database.

(eminin = 0.0) [real]
Minimum energy of the MATRIX grid in eV.

(dein = 0.5) [string]
Spacing of energy bins (in eV) in each interval of the MATRIX grid. A single value (e.g., 0.5 as the default) applies a constant grid spacing for the entire matrix. Multiple values separated by commas (e.g., '0.5,2.0') create a variable grid spacing starting at eminin. The number of channels in each energy interval must be specified with nchanin. The values may be floating point.

(nchanin = 32768) [string]
Number of channels in each interval of the MATRIX grid. Combined with the default values of the previous two parameters, the default means that the output MATRIX has 32768 channels with 0.5 eV size up to 16384 eV. If, instead, dein='0.5,2.0' and nchanin='26000,6500' the output MATRIX would have 26000 bins (with 0.5 eV size) up to 13000 eV, and 6500 bins (with 2 eV size) above 13000 eV (hence up to 26000 eV). dein and nchanin must have the same number of intervals. The values of nchanin must be integers.

(useingrd = yes) [boolean]
If useingrd is set to yes the output grid is set equal to the input grid, and the output grid parameters are ignored. In this case nchanin and dein may only have single values. If useingrd is set to no the output grid is determined by eminout, deout, and nchanout ([yes]/no).

(eminout = 0.0) [real]
Minimum energy of EBOUNDS in eV. The default value is 0.0. This parameter is ignored if useingrd is set to yes.

(deout = 0.5) [real]
Spacing of the EBOUNDS energy grid in eV. This must be consistent with the assumed energy grid of the PI file (i.e., 0.5 eV for the SXS). This parameter is ignored if useingrid is set to yes.

(nchanout = 32768) [integer]
Number of channels in EBOUNDS. This must be consistent with the number of PI channels in the PHA file. The default value for SXS is 32768. This parameter is ignored if useingrd is set to yes.

(rmfthresh = 1.0e-9) [real]
Low threshold for the RMF construction. For a given X-ray photon energy, the normalized probability (i.e., a value less than 1) of producing a certain PI value is calculated for each channel and written in the MATRIX extension. If the probability is less than this threshold, 0 (zero) is written for that element. Using the default value (1.0e-9) is secure enough for most cases.

(emincont = 10.0) [real]
Lower energy limit [eV] of electron continuum.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Create an RMF file with the default settings, using a weighting factor file.
sxsrmf infile=weightfactor.fits outfile=SXS_output.rmf

2. Create RMF and RSP files using the additional input arffile parameter.
sxsrmf infile=weightfactor.fits outfile=SXS_output.rmf outrsp=YES arffile=SXS_input.arf rspfile=SXS_output.rsp

3. Create a large-sized rmf, SXS_pixel28_Hrez.rmf, for High-resolution events for pixel 28.
sxsrmf infile=NONE outfile=SXS_pixel28_Hrez.rmf pixel=28 resol=H whichrmf=L

4. Create a small-sized (line-spread function Gaussian core only) RMF, SXS_pixel18_Mrez.rmf, for Mid-resolution events for pixel 18.
sxsrmf infile=NONE outfile=SXS_pixel18_Mrez.rmf pixel=18 resol=M whichrmf=S

5. Create a large-sized RMF, SXS_pixel0_Hrez.rmf, for High-resolution events for pixel 0, where the input grid has variable grid-spacing.
sxsrmf infile=NONE outfile=SXS_pixel0_Hrez.rmf pixel=0 eminin=0 dein=0.5,2.0 nchanin=26000,6500 useingrd=NO

NAME sxssamcnt - Calculate the local time in the SXS files necessary to assign time

USAGE sxssamcnt infile outfile

DESCRIPTION
'sxssamcnt' calculates the 'local time' and fills the column(s) SAMPLECNT(#) in the SXS event, lost GTI and HK files. Sample count represents the fine time resolution, assigned internally by the instrument electronics. After 'sxssamcnt' is run, the task 'ahtime' synchronizes the fine local time to the TT time (obtained via the GPS) written in the TIME column.

'sxssamcnt' runs : on science antico and pixel event files; on the diagnostic mode wfrb and noiserec for antico and pixel data and pulserec for the pixel data; on the gti lost for antico and pixel and on specific extensions of the HK data. The calculation of the samplecnt uses different columns depending on the input data type.
a) For all type of events detected by the pixel array the columns are: TRIG_LP, rough event trigger time, TIME_VERNIER, fine time correction, WFRB_WRITE_LP, reference for the TRIG_LP and WFRB_SAMPLE_CNT, reference for the sample cnt.
b) For the antico events the columns are : TRIG_LP, WFRB_WRITE_LP, WFRB_SAMPLE_CNT, FLG_EVENT_LOST.
c) For the HK the columns are : LATCH_SAMPLE_CNT1, reference to the sample count and LATCH_BASE_CNT1, finer clock value.
d) For the lost GTI the columns are : EL_#_LP, TRIG_LP for the start and stop, WFRB_WRITE_LP and WFRB_SAMPLE_CNT.
These column names may be set in the parameters 'col1', 'col2', 'col3', 'col4' and 'col5' depending on the input data file. The sample count calculation is written in the column(s) set by the parameter 'outcol'. For antico and HK, 'sxssamcnt' always calculates single value per each row that is written in the output column SAMPLECNT. For lost GTI, 'sxssamcnt' calculates two values for each row that are written in the columns SAMPLECNT1 and SAMPLECNT2. These are used by 'ahtime' to calculate the START and STOP.

For the pixel data, the task also calculates two local times for each row. The first, SAMPLECNTTRIG, is derived similarly to the other local time and is to enable the calculation of the event trigger time by 'ahtime'. The other, SAMPLECNT, is derived with a polynomial function of SAMPLECNTTRIG, whose coefficient are stored in CALDB, and is to enable the calculation of the event arrival time.

PARAMETERS
infile [filename]
Name of input SXS file. 'sxssamcnt' may be used on event (pixel and antico), some diagnostic science event mode SXS-specific HK, or lost GTI input files.

outfile [filename]
Name of the output file that contains the calculated sample count column(s). For antico and HK only one column SAMPLECNT is written; for lost GTI two columns are written SAMPLECNT1 and SAMPLECNT2 ; for pixel event two columns are written SAMPLECNTTRIG and SAMPLECNT.

(col1 = DEFAULT) [string]
Column name for the rough local time. If col1=DEFAULT, this is set to TRIG_LP for pixel, antico, pulserec and noiserec event files, to LATCH_Sample_Cnt1 for SXS-specific HK extension, to EL_#_LP for lost GTI, where '#' indicates that both start and stop times are processed.

(col2 = DEFAULT) [string]
Column name for the 'fine' local time correction. If col2=DEFAULT, this is set to TIME_VERNIER for pixel and pulserec event files, to LATCH_base_Cnt1 for SXS-specific HK extension. For the antico, noiserec and lost GTI files this column is ignored. For antico and lost GTI files TIME_VERNIER is set by the values in the parameters 'timever1' and 'timever2'.

(col3 = DEFAULT) [string]
Column name for the reference rough local time. If col3=DEFAULT, this is set to WFRB_WRITE_LP for pixel, antico, pulserec, noiserec and lost GTI files. For SXS-specific HK extension is ignored.

(col4 = DEFAULT) [string]
Column name for the reference sample count. If col4=DEFAULT, this is set to WFRB_SAMPLE_CNT for pixel, antico, pulserec, noiserec and lost GTI files. For SXS-specific HK extension is ignored.

(col5 = DEFAULT) [string]
Column name that specifies whether an antico event is a 'lost antico event'. If col5=DEFAULT, this is set to FLG_EVENT_LOST, and the SAMPLECNT column is set to its NULL value.

(timever1 = 1) [integer]
Time to be added to the local time calculation. For lost GTI files this value is added to the sample count corresponding to the start. For antico events this value is added to the SAMPLECNT column Default values are -8 for antico lost GTI, 0 for antico events.

(timever2 = 2) [integer]
Time to be added to the local time calculation. For lost GTI files this value is added to the sample count corresponding to the stop. Default value is 23 for antico lost GTI.

(outcol = DEFAULT) [string]
Output column name for the calculated local time. If set to DEFAULT, the name is SAMPLECNT for antico, pulserec and noiserec files, and SXS-specific HK extensions. For lost GTI files, by default the columns are SAMPLECNT1 and SAMPLECNT2 else add to '1' and '2' to the string in 'outcol'. For pixel event, by default the columns are SAMPLECNT and SAMPLECNTTRIG.

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Calculate the samplecnt for the SXS housekeeping (HK) file hk_in.fits, creating the output HK file hk_out.fits with the SAMPLECNT column populated accordingly, and using the values in the LATCH_SAMPLE_CNT1 and LATCH_BASE_CNT1 columns for the calculation.
sxssamcnt hk_in.fits hk_out.fits

2. Calculate the samplecnt for the SXS pixel event file event_in.fits, creating the output event file event_out.fits with the SAMPLECNT column populated accordingly, and using the values in the WFRB_SAMPLE_CNT, WFRB_WRITE_LP, TRIG_LP, TIME_VERNIER columns for the calculation.
sxssamcnt event_in.fits event_out.fits

3. Calculate the samplecnt for the SXS antico event file antico_in.fits, creating the output event file antico_out.fits with the SAMPLECNT column populated accordingly, using the values in the WFRB_SAMPLE_CNT, WFRB_WRITE_LP, and TRIG_LP columns for the calculation.
sxssamcnt antico_in.fits antico_out.fits

4. Calculate the start and stop samplecnt values the LOST gti file lost_in.gti, creating the output event file lost_out.gti with the SAMPLECNT1 and column SAMPLECNT2 populated accordingly, using the values in the WFRB_SAMPLE_CNT, WFRB_WRITE_LP, and TRIG_LP columns for the calculation. Extend each interval so that they extend 8 ticks earlier and 15 ticks later.
sxssamcnt lost_in.gti lost_out.gti timever1=-8 timever2=15

NAME sxsseccor -- Correct PHA for secondary events

USAGE sxsseccor infile outfile

DESCRIPTION
'sxsseccor' correct the telemetered PHA values of the medium secondary events using the time and the PHA information of all preceding events belonging to the same group. A group is defined as a primary event and all associated secondary events occurring within a pixel. The events are flags belonging to group by the task 'sxssecid' and the assignment written in the column GROUPS.

The secondary correction depends on the SXS pulse shape, which depends on energy range and pixel number. The pulse shapes are stored in a CALDB file specified by the parameter 'pulsefile'

PARAMETERS
infile [filename]
Input SXS event file name. The file must have time assigned.

outfile [filename]
Name of the output file. This is a copy of the input file with a new column added (see phaout parameter) filled with the corrected PHA values.

(pulsefile=CALDB) [filename]
Input file containing PHA pulse amplitudes for each pixel for multiple energy regions. If set to CALDB, the file is queried for in the calibration database.

(itypecol=ITYPE) [string]
Name of the column used to identify grades for the primary/secondary association and grade sequence.

(phaout=PHA2) [string]
Name of new column to fill with the corrected PHA values. An error occurs if this is set to "PHA".

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Correct PHA values for mid-resolution secondary events using the pulse amplitude file in CALDB. Event grades are determined from the ITYPE column and the corrected PHA values are written to the column PHA2.
sxsseccor infile=event_in.fits outfile=event_out.fits phaout=PHA2

NAME sxssecid -- Associate SXS secondary events to the primary and allow to recalculate event grades

USAGE sxssecid infile outfile

DESCRIPTION
'sxssecid' executes in two calculations using the SXS event file : a) recalculate the grade for each event; b) associate secondary events to the primary. The setting of the parameter 'regrade' determines the order in which sxssecid executes the two functions. If 'regrade' is set to 'yes', 'sxssecid' first recalculates the grades and after associates secondary to the primary using the new recalculated grades. If 'regrade' set to 'no', the grade are not recalculated and 'sxssecid' only associates secondary to the primary using the column specified in the

parameter 'itypecol' (d/f ITYPE). In both calculations events are excluded if : 1) they are flagged as baseline (BL) or event lost (EL) in the on-board graded column ITYPE and/or 2) any of the parameters 'ckrisetime' (event with risetime >127), 'ckctrec', 'ckctel', 'ckant' (event flagged in the STATUS column) are set to yes. By default all events below a PI threshold (see parameter 'pxpithr') are set to orphans when associate the secondary to the primary and skipped when assign new grade if regrade = yes

Regrading Calculation
There are two components in the grade definition : 1) if an event is primary or secondary and 2) if an event is High Mid or Low resolution. 'sxssecid' regrades the events per pixel by first identifying if an event is primary or secondary, using the parameter 'dtprimary', and after if it is High or Mid or Low resolution with the parameters 'dtmidhigh' and 'dtlowmid'. These parameters are used as follows. A primary event is defined as having no preceding event within 'dtprimary' and a secondary event has at least one preceding event within 'dtprimary'. An High-res event has no events within 'dtmidhigh' on either side, and Mid-res none within 'dtlowmid' on either side. Combining the two definitions : a) Hp events have no preceding event within 'dtprimary' and none within 'dtmidhigh'. b) Mp events have no preceding events within 'dtprimary' and must have the following event within 'dtmidhigh' and 'dtlowmid'. Lp events have no preceding events within 'dtprimary' and must have the following event within 'dtlowmid'. Ms events have at least one preceding event within 'dtprimary' and no event within 'dtlowmid'. Ls events have at least one preceding event within 'dtprimary' and one event within 'dtlowmid'.
The new grades are written to a new column specified by the 'itypecol' parameter using the same on-board convention which is : 0 for Hp, 1 for Mp, 2 for Ms, 3 for Lp, 4 for Ls, 5 for Baseline-BL, 6 for Lost - EL, 7 for Rejected - Rj. The name of the column may not be set to 'ITYPE' since a column named ITYPE already exist in the event file with the grades determined on on-board.

Identification of the secondary
The on-board processing assigns grades to each event however does not link the secondary to its parent primary. 'sxssecid' associates secondaries to its parent event that occurs within the same pixel. The primary-secondary association uses the grade definition as well as a timing test. The grade information is defined in a column of the event file (see parameter 'itypecol'). The timing test attempts to locate primary-secondary within the same pixel that are within a time interval specified in the parameter 'dtmidhigh'. It assumes that there is only one primary either an Hp and/or Mp and/or Lp for each pixel and all the secondaries are associated to these primary before next primary occurs.
The output of the primary-secondary association uses three columns: INDEX provides a unique index for each event, GROUPS links the secondary to the primary, SEQ records the sequence of the associated events in a group. The GROUPS column is populated as follows: Hp Mp and Ls events have GROUPS=INDEX; secondaries have GROUPS set equal to the index of the parent primary ; secondary events that do not have a primary within 'dtmidhigh' (orphans) have GROUPS set to a negative value with magnitude equal to the INDEX of the previous primary in the same pixel; events classified as baseline (BL) or lost (EL) have GROUPS set to NULL; GROUPS is set to NULL also for events that satisfy the condition set by the parameters 'ckrisetime' (event with risetime >127), 'ckctrec', 'ckctel', 'ckant' (event flagged in the STATUS column) if they are set to yes.

The sequence in the SEQ column is assigned using the ITYPE+1 on the events in that sequence. For example the sequence Mp Ms Ms is recorded as 233 where each digit is ITYPE+1. All events that belong to the same sequence have the same value in the SEQ column.
The value in SEQ is set to NULL if the event is an orphan and set to 999 if there more than 9 events in a group.

PARAMETERS

infile [filename]
Input SXS event file name. The file must have time assigned.

outfile [filename]
Name of the output file. This is a copy of the input file where the INDEX, GROUPS and SEQ columns (and the itypecol column if regrade=yes) are populated.

(itypecol=ITYPE) [string]
Name of the column used to identify grades for the primary/secondary association and grade sequence. If regrade=yes, itypecol cannot be set to ITYPE.

(dtprimary = CALDB) [string]
Time interval [ms] that distinguishes primary and secondary events. Used in the event regrading if regrade=yes. If set to CALDB, the parameter is read from the calibration database.

(dtlowmid = CALDB) [string]
Time interval [ms] that distinguishes low and mid-resolution events. Not used for primary/secondary association identification, but may be used for regrading of events if regrade=yes. If set to CALDB, the parameter is read from the calibration database.

(dtmidhigh = CALDB) [string]

Time interval [ms] that distinguishes high and mid-resolution events. Used for regrading of events if regrade=yes. If set to CALDB, the parameter is read from the calibration database.

(tol = 2.) [real]
Allowed tolerance in ns for associating secondaries and primaries.

(pxpithr = 600) [real]
If usepxpithr = ALL, events with PI below this value are labeled as orphans.

(usepxpithr = ALL) [string]
If usepxpithr = ALL, then events with PI < pxpithr are defined as orphans with no associated primary event. If regrade = yes, then such events are skipped when assigning a new grade. Events with PI below this value are labeled as orphans.

(ckctrec = no) [boolean]
If ckctrec = yes, events identified as recoil crosstalk according to the STATUS column are assigned GROUPS=NULL, and are not regraded even if regrade=yes (yes/[no]).

(ckctel = no) [boolean]
If ckctel = yes, events identified as electrical crosstalk according to the STATUS column are assigned GROUPS=NULL, and are not regraded even if regrade=yes (yes/[no]).

(ckant = no) [boolean]
If ckant = yes, events identified as coincident with antico events according to the STATUS column are assigned GROUPS=NULL, and are not regraded even if regrade=yes (yes/[no]).

(ckrisetime = yes) [boolean]
If ckrisetime = yes, events with risetime>127 are assigned GROUPS=NULL, and are not regraded even if regrade=yes ([yes]/no).

(regrade = no) [boolean]
If regrade = yes, the events are regraded based on their assigned times and the dtprimary, dtlowmid, and dtmidhigh parameters. The newly calculated grade is written into the column specified by the setting of the 'itypecol' parameter. (yes/[no]).

(buffer = -1) [integer]
Rows to buffer (-1=auto, 0=none, >0=numrows).

[ccldhm]

EXAMPLES
1. Find the associated primaries for secondaries in the file event_in.fits based on their original grades and the definition of secondaries given by the value of dtmidhigh in CALDB. Create event_out.fits with INDEX, GROUPS and SEQcolumns populated accordingly.
sxssecid infile=event_in.fits outfile=event_out.fits

2. Find the associated primaries for secondaries in the file event_in.fits based on a regrading with secondaries defined as those with preceding events within 60 ms, and high/mid/low resolution defined, respectively, as events with no events within 80 ms, at least one event within 80 (but not within 20) ms, and at least one event within 20 ms. Write the new grades to a new ITYPE2 column, and use this for identifying secondaries.
sxssecid infile=event_in.fits outfile=event_out.fits dtprimary = 60 regrade=yes dtlowmid=20 dtmidhigh=80 itypecol=ITYPE2

NAME xrtraytrace -- Perform raytracing simulations of X-ray telescopes, calculating photon paths, PSF, EEF, and effective area

USAGE xrtraytrace mirrorfile[ext] obstructfile[ext] frontreffile[ext] backreffile[ext] pcolreffile[ext] scatterfile numphoton energy transmode scattermode source offaxis roll outeafile outpsffile psfpars outphistfile

DESCRIPTION
'xrtraytrace' is a standalone raytracing code that simulates the passage of photons through a thin-foil type X-ray telescope, from the aperture through to the focal plane. Although the code is generalized to be used with any X-ray telescope of the thin-foil variety, it is currently only tested for the telescopes aboard Hitomi. Several options are possible for generating the source photons that are injected into the telescope, and details of these are described below, under the description of the input parameter 'source'. The raytracing code can be used to generate grids of results corresponding to multiple source positions relative to the telescope optical axis. The code calculates the path of each photon, and the final impact coordinates on the focal plane, if the photon survived to reach it. It also calculates the Point Spread Function (PSF), Encircled Energy Fraction (EEF), and effective area (EA). Intermediate impact coordinates along the photon

paths can also be recorded in a photon history file. This file also contains many aggregate and statistical quantities pertaining to the raytrace run. The raytracing code itself does not apply any specific instrumental masks, so the focal plane is effectively infinite in extent. Focal-plane photons must be filtered for impact on a specific instrument or focal-plane region. No detector efficiency effects are applied in the raytracing, so the effective area calculated is that for the telescope only (i.e., the effective area file is not an "ARF"). Note that the effective area includes only paths that include a "double reflection" (i.e. exactly one front-side primary mirror reflection and exactly one front-side secondary mirror reflection). The path may include other interactions such as transmission and pre-collimator reflection.

The smallest mirror and pre-collimator foil unit that the raytracing code deals with is a section of a foil that is bounded by two alignment bars in the support structure, and these alignment bars define a "sector." The smallest foil unit is named sub-foil, and each sub-foil corresponds to a unique row in the FITS Telescope Description File (TDF), whose columns define the sub-foil properties.

The coordinate system used by 'xrtraytrace' is inherited directly from the TDF. In this system the positions of the mirror foils are fixed by describing a point on a foil by its radius from a central axis, and its height from the focal plane. The focal plane is the x-y plane and the z-axis is the central axis of the cylindrical telescope structure. This Cartesian frame of reference is used for all telescope components and for all photon positions and direction vectors. When misalignments are applied to the various telescope components, the abstract coordinate system remains fixed in that the x-y plane is always the focal plane, and the z-axis is always the original central telescope axis. All rotations and/or shifts are defined relative to this fixed coordinate system.

NOTE: The mirror calibration file and the 'xrtraytrace' were changed to account of any rotation of the telescope respect to the focal plane in Dec 2016. This informaton is stored in a keyword, TELFPROT that is read by 'xrtraytrace'. Previosu version of the software do not read this keyword and the results do not account for the rotation.

INPUT FILES

Required:
Telescope Description File (TDF)
Examples:
ah_sxs_mirror_20131001v002.fits (SXT-S)
ah_hx1_mirror_20131001v002.fits (HXT1)
The TDF contains information the full geometrical structure of the telescope, as well as details of the properties of the reflective surface coatings of the mirror foils (see parameter 'mirrorfile').
Note that "old-style" TDF's (such as those for ASCA and Suzaku) do not work with the current code because the file format has evolved to become more complex.
Reflectivity/Transmission file
Examples:
ah_sxs_reftrans_20131001v002.fits
ah_hx1_reftrans_20131001v002.fits
This file describes the reflectivity and transmission properties of the telescope mirror and pre-collimator foils as follows (see parameter 'frontreffile'):
1st Extension: Front-side mirror reflectivity and surface thin-film transmission
2nd Extension: Mass-absorption coefficients of all the materials making up the surface and bodies of mirror and pre-collimator foils.
3rd Extension: Reflectivity of the back-sides of mirror foils.
4th Extension: Reflectivity of the pre-collimator foils.
(Note that 'xrtraytrace' calculates the net transmission from the front-side to the back-side of foils using BOTH the transmission data in the 1st extension and the mass-absorption coefficient data in the second extension.)

The file is generated by initially running the tool 'xrtreftable', which uses the TDF to generate the 1st and second extension. The third and fourth extension data cannot be calculated from theory and are based on measurements. The third and fourth extensions must be appended onto the file generated by 'xrtreftable'.
The raytracing code can handle both eV and keV units of energy in the reflectivity file extensions.

Optional:
Energy grid file
One method for specifying the photon energies for 'xrtraytrace' is to use a FITS file that contains a column that is an energy grid. (See description of the input parameter 'energy' for the other methods of specifying photon energy.)
The name of the file and extension containing the energy grid are specified as a single parameter string for the input parameter 'energy'.

Photon source file
Photons that are to be raytraced can be generated either by 'xrtraytrace' (for which there are several options), or their properties can be read from a file. See the description of the input parameter 'psrcfile' below for details on the expected file format for the two types of photon file input that are supported.

OUTPUT FILES
All of the output files are optional. However if no output files are requested, 'xrtraytrace' aborts.
1. PSF (Point Spread Function) or EEF (Encircled Energy Fraction) file. The 'xrtraytrace' code writes either a PSF image file OR an EEF file, depending on input parameter settings (see descriptions for the input parameters 'psfpars' and 'outpsffile' below).
If there are 10 or fewer energy points in the input energy grid, a PSF image or EEF function is written to a separate extension for each energy, off-axis angle, and azimuthal (roll) angle. However, if more than 10 energies are in the energy list, then the PSF or EEF is summed over all photon energies for a given pair of off-axis and azimuthal angles. A unique extension is written for each unique pair of angles.

The PSF image is normalized so that the sum over all photons in the focal plane is 1.0. The EEF is normalized so that it is 1.0 at the radial position on the focal plane of the furthest photon impact from the optical axis.
The PSF image always shows the source at the center of the image. The actual position of the center of the source can be found in the FITS header keywords XCENTER and YCENTER.

2. Effective Area (EA) file, containing the effective area (of the telescope only), as a function of energy (see description for the input parameter 'outeafile' below for details).

3. Photon history file. Records path history information for each photon, as well as aggregate and statistical quantities for the entire raytrace run. See description for the input parameter 'outphistfile' for more details about the history file contents, and caveats.

PARAMETERS
mirrorfile [filename]
Name of the telescope description file (TDF) and the extension that holds the geometrical description of primary and secondary mirror foils (e.g. MIRROR). It is assumed that the name of the pre-collimator extension is COLLIMATOR. If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

obstructfile [filename]
Name of the telescope description file and the extension that holds the geometrical description of the telescope support structures (e.g. OBSTRUCT). If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

frontreffile [filename]
Name of the reflectivity file and the extension for the front-side reflectivity of the mirror foils. The extension also includes the thin surface film transmission. The names of the reflectivity and transmission columns are linked to groups of mirror foils that they apply to, by means of a column called FREFLECT in the TDF. The second extension in the reflectivity file contains mass-absorption coefficients that are used by xrtraytrace to calculate transmission probabilities of the "thick" materials (as opposed to thin films) in the telescope. If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

backreffile [filename]
Name of the reflectivity file and the extension for the back-side reflectivity of the mirror foils. If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

reffilepcol [filename]
Name of the reflectivity file and the extension for the reflectivity of the pre-collimator blades/foils. The pre-collimator reflectivity is the same for front-side and back-side reflection. If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

scatterfile [filename]
Name of the file containing the scattering angle probability distributions for the direction of reflected rays relative to the specular direction. The file contains data for the front-side of mirror foils, the back-side of mirror foils, and for the pre-collimator blades. In general, foils in different physical regions of the telescope can have different scattering distributions. The column names are referenced in the SCATTER column in the TDF. This field is only used if the "scattermode" parameter is not 'none'. If CALDB is specified, the file is read from the calibration database, based on the TELESCOP and INSTRUME input parameters.

numphoton [integer]
For running modes that do not take input photons from a file (see parameter 'source'), 'numphoton' is the number of input photons entering the telescope aperture for each unique value of the energy (see 'energy' parameter), and off-axis angle and azimuthal angle pairs. For running modes that do take input photons from a file, each row of the input file corresponds to one photon.

energy [string]
In the case that the running mode does not take input photons from a file (see parameter 'source'), the 'energy' parameter specifies the energies of the input photons in a number of different ways as follows: If the string 'energy' consists of a series of numbers, and if the

first number is equal to -1, then the energy grid for input photons is taken from the energy grid of the reflectivity file, between lower and upper boundaries that are specified by the second and third number in the string. If the string 'energy' consists of a series of numbers, and if the numbers are all positive, then the energy grid for input photons is taken to be the list of all the numbers in the string specified for 'energy', in units of keV. If the string 'energy' is not a series of numbers, it is interpreted as a FITS filename and extension. The energy grid is read from a column called "ENERGY" from the extension that is included in the input parameter string. The units of the energy in the file must be keV or eV. For raytrace runs that use a multilayer reflectivity file (e.g. for HXT1 or HXT2), all of the raytracing energies in the raytracing energy grid must have exact values that are present in the energy grid of the reflectivity file. With option (1) above this is automatically satisfied but, for the other two options, this correspondence must be explicitly satisfied by the input list (option 2) or file (option 3). If there is a mismatch in the energy grids and the reflectivity file describes a multilayer mirror, the raytracing program stops because correct interpolation of the reflectivity cannot be guaranteed.
For running modes that do take input photons from a file, there are two options for the photon energies. First, if the keyword UNQELIST in the input photon FITS file is TRUE, then an energy grid of unique values is read from the list of numbers specified in the string 'energy'. This unique list of energies is used to pre-interpolate reflectivity and transmission values (to reduce runtime). Energy values in each row of the input photon file are then compared with the unique energy list. The second option is to use the energy values in each row of the energy column of the input photon file (one row corresponds to one photon), even if it means interpolating the reflectivity and transmission more than once for the same energy.

(seed = 0) [integer]
Random number seed. If the value is zero then the code uses the system time to generate a random number.

(misalign = "1 1 1 1 1 1") [string]
This parameter, a string of 6 integers, controls turning on and off the 'tilt' and 'twist' angular misalignment offsets in the TDF columns SYSTILT and SYSTWIST. The six numbers correspond to the following: (1) tilt for pre-collimator, (2) tilt for primary mirrors, (3) tilt for secondary mirrors, (4)
twist for pre-collimator, (5) twist for primary mirrors, (6) twist for secondary mirrors.

A negative value of any of the six numbers turns off the corresponding misalignment component for all sub-foils in the stated category. The 'tilt' is an angular offset from the default foil position (in arcsec, a positive value specifying a rotation of the top of the foil towards the telescope symmetry axis). The 'tilt' rotation requires a pivot axis and three such axes are defined for the pre-collimator, primary, and secondary mirror foils. A single number between 0.0 and 1.0 for each of the 3 categories of foil specifies the fractional distance of the axis from the bottom edge of a sub-foil, the axis being parallel to the bottom edge of the sub-foil. The keywords in the TDF are as follows:
TLTPVPCL : for pre-collimators, in the COLLIMATOR extension of the TDF
TLTPVPRI : for primary mirrors, in the MIRROR extension of the TDF
TLTPVSEC : for secondary mirrors, in the MIRROR extension of the TDF
The 'twist' angular offset (in arcsec) is a rotation around an axis that is perpendicular to the tilt axis, intersecting it at the midpoint, and parallel to the optical axis of the telescope.
The 'misalign' parameter set should not be changed from the default values to maintain fidelity with the calibration of the telescope as embodied in the TDF. The only reason to adjust the misalign parameter set would be for fine-tuning the calibration.
transmode [string, (NONE/ALL/MIRROR/PCOL)]
Method for treating transmission of photons in the mirror and pre-collimator.
none: Assume transmission is zero for all mirror and pre-collimator foils.
all: Calculate transmission of photons for both mirror and pre-collimator foils.
mirror: Calculate transmission of photons only for mirror foils, force transmission to be zero for pre-collimator foils.
pcol: Calculate transmission of photons only for pre-collimator foils, force transmission to be zero for mirror foils.

scattermode [string, (NONE|ALL|MIRROR|PCOL)]
Scattering "switch" for treating scattering of photons on mirror and pre-collimator foils. If this field is not 'none', there must be a valid file in the "scatterfile" field.
none: No scattering for either mirror or pre-collimator foils, all reflection is purely specular.
all: Treat scattering for both mirror and pre-collimator foils.
mirror: Treat scattering only for mirror foils, assume specular reflection for pre-collimator foils.
pcol: Treat scattering only for pre-collimator, assume specular reflection for mirror foils.

source [string, (POINT|FLATCIRCLE|BETAMODEL|PHOTONLIST|GROUNDMODEL|DIAGNOSTICMODE)]
Method for generating input photons for the raytracing.
point: Point source at infinity. Photons arrive at random points on the active region of the telescope aperture.
flatcircle: Extended source at infinity that has a spatial distribution that has uniform flux over a circular region (zero outside of the circle).
betamodel: Extended source at infinity that has a spatial distribution described by the "beta model" (see 'betapars' parameter).
photonlist: Read input photon energy and direction from the FITS file specified by the input parameter 'psrcfile' (see below for details). The source is at infinity and need not be a point source. Each row in the input file corresponds to one photon.

groundmodel: Read input photon energy, direction, and three-dimensional position of the photon origin point, from the FITS file specified by the input parameter 'psrcfile' (see below for details). The source is at a finite distance defined by the position of the origin of source X-rays in each row of the input FITS file. This mode is useful for ground calibration.
diagnosticmode: Inject photons at a single entry point on the telescope aperture. The parameters defining the entry position are specified by the input parameter string 'diagpars' (see below for details).

(betapars = "0.50 0.60 5.0") [string]
Parameters of the beta model if 'source=beta' is specified.
1st number in betapars: beta model core radius in arcmin.
2nd number in betapars: the index "beta" of the beta model.
3rd number in betapars: the maximum radius (in arcmin) of the source spatial distribution.

(flatradius = 10.0) [real]
The radius (in arcmin), of the extended source for the flat spatial distribution option, 'source=flatcircle'.

(psrcfile) [filename]
Name of the input photon file and extension for the options 'source=photonlist' and 'source=groundmodel'. In both cases the extension containing the data must be specified as part of the input file name. Each row of the file corresponds to one photon. The columns describe various attributes of the input photons. Note that the telescope coordinate system is such that the x-y plane coincides with the focal plane, and the z-axis corresponds to the telescope rotational symmetry axis.

For 'source=photonlist'
The columns of the FITS file are:
Energy (keV)
Off-axis angle (relative to the optical axis) of the photon origin (radians)
Azimuthal angle made by the photon direction in radians, relative to the x-axis.
For 'source=groundmodel':
The columns of the FITS file are:
Energy (keV)
Initial x-position of the photon (mm)
Initial y-position of the photon (mm)
Initial z-position of the photon (mm)
x-component of unit vector corresponding to the photon direction
y-component of unit vector corresponding to the photon direction
z-component of unit vector corresponding to the photon direction.
For both the 'source=photonlist' and 'source=groundmodel' options, the file 'psrcfile' must have the following keywords:
UNQELIST (boolean): "True" if the user intends to specify (in the raytracing parameter 'energy'), the unique list of energies that occur over all rows of the file 'prscfile.' If "False" the raytracing code interpolates reflectivity and transmission grids for every row of 'psrcfile', even if it means repeating the interpolation for the same energy. The runtime for UNQELIST=F may be significantly greater than the run time with UNQELIST=T.
MINENERG (real): Minimum energy amongst all rows of the file 'psrcfile'
MAXENERG (real): Maximum energy amongst all rows of the file 'psrcfile'

(diagpars = "2 5 174 0.70 0.50") [string]
A string containing parameters that define the entry point on the telescope aperture for single-point injection for 'source=diagnosticmode'. The meanings of the numbers are as follows:
1st number: (a) The telescope segment number in which the entry point is located, or (b) any negative number, meaning that the user enters the initial x and y in the fourth and fifth numbers.
2nd number: The sector number in which the entry point is located. If the 1st number is negative, this number is ignored.
3rd number: The shell number closest to the telescope optical axis of the pair of shells between which the photon entry point is located. If the 1st number is negative, this number is ignored.
4th number: The radial location of the entry point in terms of the fraction of the radial gap between the outer radius of the foil in the lower shell, and the outer radius of the foil in the upper shell. If the 1st number is negative, this number is instead the desired direct initial photon x-coordinate.
5th number: The location of the entry point in terms of the fraction of the angle between the boundaries of the sector defined by the second number. If the 1st number is negative, this number is instead the desired initial photon y-coordinate.

offaxis [string]
For running modes that do not take input photons from a file, the list of numbers in the string 'offaxis' corresponds to off-axis input photon source positions (angular deviation from the telescope optical axis in arcmin).
For extended sources, the off-axis angle corresponds to the offset of the center of the source.

For each off-axis angle, the raytracing code injects 'numphoton' photons into the telescope for EACH energy in the specified energy grid. For each off-axis angle and energy pair, the roll angle can take a single value, or it can take several values in a nested loop (see parameter 'roll' below).

For each triple set of numbers consisting of energy, off-axis angle (offaxis), and roll angle, xrtraytrace injects 'numphoton' photons.

The raytraced events appears in a single output file regardless of the photon energy, off-axis angle, and roll angle (see description of the input parameter 'outphistfile' below).

roll [string]

For running modes that do not take input photons from a file, the list of numbers in the string 'roll' corresponds to input photon source directions that have the same angular offset (parameter 'offaxis') from the optical axis, but the photon direction vector makes different rotation angles relative to the x-axis. The units of the azimuthal, or 'roll', angle are degrees.

For the extended source options 'flatcircle' and 'betamodel', the roll angle corresponds to the rotational offset of the direction of the ray to the center of the source.

For each triple set of numbers consisting of energy, off-axis angle (offaxis), and roll angle, xrtraytrace injects 'numphoton' photons.

There are two modes of operation:

(a) Normal Mode: If all of the numbers in the 'roll' list are positive, then a raytracing run is performed for each off-axis angle for every value of roll angle in the list (i.e., by means of nested loop, the total number of runs is equal to the number of off-axis angles multiplied by the number of roll angles).

(b) Pair Mode: If the first value of roll in the list is negative then for each value of off-axis angle, the value of the roll angle is taken from the corresponding position in the list (i.e. the nth off-axis angle run is performed for only the nth roll angle). In the latter mode the total number of runs is simply equal to the number of off-axis angles. The values of roll angle used are the absolute values of the list members. The code checks that the number of roll angle values is exactly equal to the number of off-axis angle values. If this is not the case, the program stops.

If xrtraytrace is to be run in "Pair Mode" and the first roll angle desired in the list is 0.0, the actual value that should be used is -360.0, because -0.0 is not interpreted correctly.

The actual roll angle used for any of the numbers in the 'roll' parameter list is the modulus of the absolute value of that number, with 360 degrees. Thus, one should never require any of the roll angles for actual input to a raytrace run to be negative; a genuinely negative roll angle should have 360 degrees added to it before putting it in the list.

The raytraced events appears in a single output file regardless of the photon energy, off-axis angle, and roll angle (see description for the input parameter 'outphistfile' below).

The roll angle in xrtraytrace is not the same quantity as the satellite roll angle and the two should not be confused with each other.

(annulus = "-1 100000.0 0.0 360.0") [string]

This parameter defines a partial annular region on the telescope aperture that effectively restricts the entry of photons relative to the full aperture. It is used to compare raytracing simulations with ground-based experimental results since a telescope generally does not have its aperture restricted in this way in-flight. The four numbers in the input parameter string 'annulus' are defined as follows:

1st number: Either (a) Minimum radius of partial annulus (mm) or (b) Negative value, in which case xrtraytrace uses the inner housing radius for the inner radius of an annular aperture. Note that the inner radius of the aperture is allowed to go to zero, in which case photons entering at smaller radii than the inner housing radius impacts the cover of the central "hole" in the telescope.

2nd number: Either (a) Maximum radius of partial annulus (mm) or (b) Negative value in which case xrtraytrace does not use an annulus for the aperture, and attempts to use the parameters of a rectangular aperture (however, if the rectangular option is turned off, an annular aperture is used by default).

3rd number: Start angle of partial annulus relative to the x-axis (degrees)

4th number: End angle of partial annulus relative to the x-axis (degrees)

If the upper radius of the annulus is greater than the outer housing radius, (as defined by the keyword PMAXRAD in the TDF), the outer radius is internally set equal to the outer housing radius by the raytracing code.

If a rectangular aperture is requested, then the outer annulus radius should be set to a negative value. (See description for the input parameter 'rectangle.') Note that both the partial annulus and rectangular aperture can be turned off in this way, in which case the full telescope aperture is used (which is an annulus).

(rectangle = "0.0 0.0 -40.0 -50.0") [string]

This parameter defines a rectangular region on the telescope aperture that effectively restricts the entry of photons relative to the full aperture. It is used to compare raytracing simulations with ground-based experimental results since a telescope generally does not have its aperture restricted in this way in-flight. The four numbers in the input parameter string 'annulus' are defined as follows:

1st number: X-coordinate of center of rectangle (mm)

2nd number: Y-coordinate of center of rectangle (mm)

3th number: Width (x-dimension) of rectangle (mm)

4rd number: Height (y-dimension) of rectangle (mm)

If any corner of the rectangle does not lie inside the annulus that corresponds to the full telescope aperture, then the raytracing code simply uses the full annulus-shaped aperture.

The rectangular aperture can be switched off by setting either (or both) of the values of the rectangle height and width to have a negative value. If the rectangular aperture is turned off, an annular aperture is used with the parameters defined by the input parameter string 'annulus' if both the inner and outer annulus radii have positive values, otherwise the full annular aperture is used.

outeafile [filename]
Name of the output effective area (EA) FITS file. It is only written if the input photons to the raytracing code are not from a file. Each extension contains the effective area versus energy function for each pair of values of the off-axis angle and the roll angle for which the raytracing run was performed.
Only photon paths that include exactly one primary mirror reflection and exactly one secondary mirror reflection (i.e. a "double reflection") contribute to the EA. The path may or may not include reflection on the pre-collimator surfaces, back-side mirror surfaces, and transmission events. Paths that do not include a double reflection are likely to appear as noise or background in real data and do not in general contribute to the principal focused image.
The EA file can be suppressed by specifying "none" for the file name.

outpsffile [string]
Name of the output file containing the PSF (point-spread function) images or the EEF (encircled energy fraction). It is only written if the input photons to the raytracing code are not from a file. Two types of file can written:
Type 1: An image in each extension corresponding to a pair of off-axis and roll angles, and each energy, if there are 10 or fewer energies in the input energy grid. If there are more than 10 energies, the PSF is summed over all energies for each pair of offset angles.
Type 2: An EEF (encircled energy fraction) versus radial distance from the central peak (in arcsec) in each extension corresponding to a unique pair of off-axis and roll angle values and a unique energy if there are 10 or fewer energies in the input energy grid. If there are more than 10 energies, the EEF is constructed using all energies for each pair of offset angles.
The PSF/EEF file can be suppressed by specifying "none" for the file name.
psfpars [string]
Numbers in the input parameter string specify parameters for the PSF or EEF calculation as follows:
1st number: (integer) 1= Type 1 file, PSF images; 2=Type 2 file, EEF functions
2nd number: (integer) For type 1 files, this is the number of pixels on the X and Y axes are 2N+1 for both X and Y, where N=second number in the string psfpars. For type 2 files this number is equal to the number of radial bins centered on the peak of the photon distribution on the focal plane.
3rd number: For type 1 files this number is the size of the pixels in the PSF image, in arcsec. For type 2 files this number is equal to the size of the radial bins for the EEF, in units of arsec.

(resultsplanez = 0.0) [real]
The z-coordinate (mm) of the x-y plane that are recorded as part of the final impact coordinates for photons that successfully emerge from the telescope pre-collimator, mirror, and support structure without being absorbed. The default is 0.0, which means that this "results plane" is the focal plane.

(resplaneonly = no) [boolean]
Write results to the photon history file only for those photons that impact the results plane (each row if the history file event data corresponds to one photon).

outphistfile [filename]
Name of the output photon history (event) file. This is a FITS file that contains detailed information about the photons that were traced and their paths.
The history file can be suppressed by specifying "none" for the file name.

The history file has two extensions:

Extension 1 (PHOTONHISTORY) contains between 12 and 45 columns, as described below.
Extension 2 (INPUTPHOTONS) contains 4 columns as follows:
Column 1 = INITIALTHETA: off-axis angle of the incident photon direction vector.
Column 2 = INITIALAZIMDIR: azimuthal angle of the incident photon direction vector.
Column 3 = ENERGY: photon energy (keV).
Column 4 = NUMPHOTONS: number of input photons for a given INITIALTHETA, INITIALAZIMDIR, and ENERGY.
The photon direction vector is referenced to the telescope coordinate frame, in which the z-axis coincides with the optical axis, and the focal plane coincides with the x-y plane. The history file contains many quantities in FITS header keywords that are aggregates or statistical constructions of useful information about the raytraced photon path histories through the telescope.

There are three options for the amount of detail that is written to the PHOTONHISTORY extension, selectable with the 'phisttype' parameter described below. Each row of data in the history file corresponds to one photon that was input to the raytracing code. Depending on what was selected for the parameter 'resplaneonly', the extension can contain information about every photon that was input to the raytracing code, regardless of the eventual fate of that photon, or it can contain information on only those photons that impacted the results plane (see parameter description for 'resultsplanez' for definition of the results plane.

The history file contains up to 45 columns. If all 45 columns are selected to be written ('phisttype=full') there are 21 columns corresponding to the x, y, and z coordinates of up to 7 interactions between the photon corresponding to a particular row, and the telescope components. The 7 sets of coordinates do not include the initial position or the final position (these are contained in separate columns).

Note that, for events that do not reach the focal plane, the final impact coordinates are not always calculated by the code in the interest of keeping the runtime as low as possible. Any coordinates in the history file columns that are not computed by the code are given the dummy value of -1.0e30, or similarly large negative number.

Note also that there are no columns giving the initial z-coordinates of the photons because, except when the source is not at infinity, they all have a single value corresponding to the z-coordinate of the aperture, which is given by the FITS header keyword, ZAPRTURE in the history file. (In the case of the 'groundmodel' option, the initial z-coordinates of the photons are specified individually and they are already in the photon list input file that must be supplied for the 'groundmodel' option.)

The 45 columns in the history file are as follows:

energy (1D, double) Input photon energy (keV)

initialrad (1D, double) original radial position (in mm) of the incident photon impact location on the telescope aperture, relative to the optical axis, in a plane perpendicular to the optical axis (i.e. parallel to the x-y plane).

initialazimpos (1D, double) Original rotational angle (or azimuthal angle), in degrees, of the incident photon impact location on the telescope aperture.

initialx (1D, double) Original telescope x coordinate (in mm) of the position of the incident photon on the telescope aperture.

initialy (1D, double) Original telescope y coordinate (in mm) of the position of the incident photon on the telescope aperture.

initialtheta (1D, double) Original off-axis angle (in arcmin) of the source position vector with respect to the telescope optical axis.

initialazimdir (1D, double) Original rotational angle (in degrees) of the source position vector with respect to the x-axis in the plane perpendicular to the optical axis.

finalxpos (1D, double) Photon impact x-coordinate on the results/focal plane (in mm) relative to the origin. Photons that do not make it to the results/focal plane can be filtered using the value of the Boolean variable in the "resultsplane" column.

finalypos (1D, double) Photon impact y-coordinate on the results/focal plane (in mm) relative to the origin. Photons that do not make it to the results/focal plane can be filtered using the value of the Boolean variable in the "resultsplane" column.

finalxpsf (1D, double) Photon impact x-coordinate on the results/focal plane relative to the source center position on the focal plane. Photons that do not make it to the results/focal plane can be filtered using the value of the Boolean variable in the "resultsplane" column. The finalxpsf coordinate has the source at the origin (finalxpos is the actual x-coordinate).

finalypsf (1D, double) Photon impact y-coordinate on the results/focal plane relative to the source center position on the results/focal plane. Photons that do not make it to the results/focal plane can be filtered using the value of the Boolean variable in the "resultsplane" column. The finalypsf coordinate has the source at the origin (finalypos is the actual y-coordinate).

pathcode (32A, string) Photon path history coded as a single 32-character string per photon. Each group of four characters corresponds to an integer that contains encoded information about an interaction between the photon and a telescope component. Up to eight events are recorded, starting with the first interaction. If there were more than 8 interactions including the final event, only the first 8 interactions are recorded.

Designating the 4 characters of the "pathcode" portion for each interaction as ABCD, the definitions are as follows:

A=(interaction type)
1=absorption, or end of path on the results (or focal) plane
2=reflection not followed by scattering
3=reflection followed by scattering
4=transmission
9=photon path incurred an error or anomalous condition
BC=(impacted object)
01=results plane (default=focal plane)
02=inner housing
03=outer housing
04=segment/sector side wall
05=pre-collimator foil
06=primary mirror foil
07=secondary mirror foil
08=support structure

09=top external object (e.g. top thermal shield)
10=bottom external object (e.g. bottom thermal shield)
D=(impacted face)
1: mirror, pre-collimator foils: back (facing outer housing)
obstructions: top
results/focal plane: top
2: mirror, pre-collimator foils: front (facing optical axis)
obstructions: bottom (unlikely)
3: mirror, pre-collimator foils: top face/edge
4: mirror, pre-collimator foils: bottom face/edge
5: mirror, pre-collimator foils: sides
6: Undetermined (but neither back nor front)

radialdir (1D, double) In cylindrical coordinates, this is the radial component of the unit vector of the radial direction of the photon if it impacts the results/focal plane. For photons that do not make it to the results/focal plane, it is the value at the last interaction (i.e. the last absorption event)

azimuthdir (1D, double) In cylindrical coordinates, this is the azimuthal angle made by the unit vector corresponding to the direction of the photon if it impacts the results/focal plane. For photons that do not make it to the results/focal plane, it is the value at the last interaction (i.e. the last absorption event).

finalxdir (1D, double) In Cartesian coordinates, this is the x-component of the unit vector of the direction of a photon if it impacts the results/focal plane. For photons that do not make it to the results/focal plane, it is the value at the last interaction (i.e. the last absorption event).

finalydir (1D, double) In Cartesian coordinates, this is the y-component of the unit vector of the direction of a photon if it impacts the results/focal plane. For photons that do not make it to the results/focal plane, it is the value at the last interaction (i.e. the last absorption event).

finalzdir (1D, double) In Cartesian coordinates, this is the z-component of the unit vector of the direction of a photon if it impacts the focal plane. For photons that do not make it to the results/focal plane, it is the value at the last interaction (i.e. the last absorption event).

numint (1J, integer) Total number of interactions between the photon and X-ray telescope components, including results/focal-plane impact if there was one.

The content of this column depends on the value of the parameter 'resplaneonly.'

If resplaneonly=no then the column is:

resultsplane (Boolean/string) True/T if the photon's path ended with an impact on the results plane (focal plane by default).

If 'resplaneonly=yes' then the column is:

rowindex (1J, integer) The value in each row is equal to the row number this photon would be in if xrtraytrace had been run with resplaneonly=no. In other words, it is the photon number in the raytracing sequence that counts all raytraced photons regardless of whether they impacted the results plane (focal plane by default) or not.

errorcode (1J, integer) An integer indicating any anomalous situations or error conditions encountered by the photon anywhere along the path. A value of zero means that no errors or anomalous situations were encountered. Currently the only error code is 999, covering all anomalies. Future versions of the raytracing code will have a larger number of error codes that distinguish between a variety of anomalous scenarios.

event1x (1D, double): x-coordinate of 1st interaction
event1y (1D, double): y-coordinate of 1st interaction
event1z (1D, double): z-coordinate of 1st interaction
event2x (1D, double): x-coordinate of 2nd interaction
event2y (1D, double): y-coordinate of 2nd interaction
event2z (1D, double): z-coordinate of 2nd interaction
event3x (1D, double): x-coordinate of 3rd interaction
event3y (1D, double): y-coordinate of 3rd interaction
event3z (1D, double): z-coordinate of 3rd interaction
event4x (1D, double): x-coordinate of 4th interaction
event4y (1D, double): y-coordinate of 4th interaction
event4z (1D, double): z-coordinate of 4th interaction
event5x (1D, double): x-coordinate of 5th interaction
event5y (1D, double): y-coordinate of 5th interaction
event5z (1D, double): z-coordinate of 5th interaction
event6x (1D, double): x-coordinate of 6th interaction
event6y (1D, double): y-coordinate of 6th interaction
event6z (1D, double): z-coordinate of 6th interaction
event7x (1D, double): x-coordinate of 7th interaction
event7y (1D, double): y-coordinate of 7th interaction
event7z (1D, double): z-coordinate of 7th interaction

priobjid (1J, integer) ID number of the first front-side primary mirror impacted. If no primary mirrors were impacted by a photon in a particular row, the value is set to -1.

secobjid (1J, integer) ID number of the first front-side secondary mirror impacted. If no secondary mirrors were impacted by a photon in a particular row, the value is set to -1.
priincangle (1D, double) Grazing incidence angle for the first front-side primary mirror impacted. If no primary mirrors were impacted by a photon in a particular row, the value is set to 0.0
secincangle (1D, double) Grazing incidence angle for the first front-side secondary mirror impacted. If no secondary mirrors were impacted by a photon in a particular row, the value is set to 0.0
(phisttype = full) [string, (full|basic|brief)]
If 'phisttype= full', all 45 columns listed for outphistfile are written to the history file.
If 'phisttype = basic', the same columns are written as for "full", except that the 21 columns for the photon path coordinates (event1x, etc.) are not written.
If 'phisttype = brief', only the following 12 columns are written (see original descriptions from the full set):
ENERGY (keV)
INITIALX
INITIALY
INITIALTHETA
INITIALAZIMDIR
FINALXPOS
FINALYPOS
PATHCODE
FINALXDIR
FINALYDIR
FINALZDIR
RESULTSPLANE (if 'resplaneonly=no')
or
ROWINDEX (if 'resplaneonly=yes')
These columns have been described above.

(externobjects = all) [string, (none|all|top|bottom)]
Treatment of objects that are situated above and below the telescope body that are described in the AZIMUTHALSTRUCT extension of the TDF (for example, thermal shields and central cover).
"all" means include both top and bottom objects in the raytracing
"top" means include only those that lie above the main telescope
"bottom" means include only those that lie below the main telescope
"none" means exclude all external objects

(fastmode = yes) [boolean]
If 'fastmode=yes', the raytracing code runs a factor of a few faster (depending on input parameters) than if 'fastmode=no'. However, with 'fastmode=yes', the results may not be accurate when the off-axis angle is large, the transmission significant, or misalignments severe. The 'fastmode=yes' option is intended for a fast calculation of effective area for nominal pointing positions.

(validdate = 2999-01-01) [string]
The UTC date (in yyyy-mm-dd format) when this calibration data should first be used. This is the date as of which the xrtraytrace output files are valid. This date is written to all output files in the keyword CVSD0001, which is required for any file entered into CALDB.

(validtime = 00:00:00) [string]
The UTC time (in hh:mm:ss format) on the day 'validdate' when this calibration data should first be used. This is the time as of which the xrtraytrace output files are valid. This date is written to all output files in the keyword CVST0001, which is required for any file entered into CALDB.

[ccldhm]

EXAMPLES
1. The following example runs xrtraytrace for the case of the Hitomi SXT with a point source at infinity, on-axis, full aperture illumination, 5 keV photon energy, and other options as detailed in the parameter list:

```
xrtraytrace mirrorfile="ah_sxs_mirror_20131001v002.fits[MIRROR]" obstructfile="ah_sxs_mirror_20131001v002.fits[OBSTRUCT]" \
frontreffile="ah_sxs_reftrans_20131001v002.fits[AH_SXT_FRONT]" \
backreffile="ah_sxs_reftrans_20131001v002.fits[REFPROBBACK]" \
pcolreffile="ah_sxs_reftrans_20131001v002.fits[REFPROBPCOL]" scatterfile="ah_sxs_scatter_20131001v002.fits" \
numphoton=10000 energy="5.0" seed=29075 misalign="0 0 0 0 0 0" transmode="none" scattermode="none" source="point" \
betapars="0.50 0.60 5" flatradius=5.0 psrcfile="none" diagpars="2 5 174 0.70 0.50" offaxis="0.0" roll="0.0" \
```

annulus="-1 100000.0 0.0 360.0" rectangle="0.0 0.0 -40.0 -50.0" outeafile="xrtraytrace_ea.fits" outpsffile="xrtraytrace_psf_img.fits" \
psfpars="1 100 0.25" resultsplanez=0.0 outphistfile="xrtraytrace_history.fits" validdate="2999-01-01" \
validtime="00:00:00" clobber="no" chatter=2 logfile="xrtraytrace.log" debug="no" history="yes" mode="ql"

The example run produces the output files:
xrtraytrace_ea.fits
xrtraytrace_psf_img.fits
xrtraytrace_history.fits
xrtraytrace.log

NAME xrtreftable - Calculates the probability of reflection and transmission of X-ray photons impinging on a reflecting surface, as a function of energy and incident angle. Also calculates mass-absorption coefficients of constituents of the reflecting foils.

USAGE xrtreftable telescop instrume atomicfile atmscafile atmsctng energyfile anglefile mirrorfile outfile outext roughmodel desc

DESCRIPTION
'xrtreftable' calculates the probability of reflection (or reflectivity) and transmission of a photon impinging on a reflecting surface. The reflection and transmission depend on the energy of the photon and incident angle. The tool obtains information on the composition of the mirror surface from an extension called SURFACE in the telescope description file (TDF). Both single-layer and multi-layer coatings for the mirrors are supported by 'xrtreftable'. The reflection and transmission probabilities are written to the first extension of the output file, as a function of energy and incident angle. The tool supports more than one group of mirror foils, whereby the mirrors in the same group share the same surface coating properties (for example, in a real X-ray telescope, mirror groups having different surface compositions may be arranged to be located at different radii ranges). The tool 'xrtreftable' writes the reflectivity of one mirror group as one column, and the corresponding transmission for that group as another column with each column containing a two-dimensional array of probability as a function of energy and incident angle.

In raytracing applications, transmission through the "thick" components of mirror foils (as opposed to the "thin" surface) may also be required, so 'xrtreftable' calculates mass-absorption coefficients as a function of energy and these are stored in extension 2 of the output file, one column per "thick material." The thick material properties are also specified in the SURFACE extension of the TDF.

OUTPUT

The first extension of the output FITS file contains the reflection and transmission probabilities as a function of energy and incident angle. The first column is a repeat of the energy grid given in the input file. There are reflection and transmission columns for each group from the mirror surface file. These columns are vectors, with one element per angle. Thus, if there are 800 energies and 600 angles, then the output extension have 800 rows, and each reflection and transmission column is a vector with 600 elements.

The second extension in the output file holds the mass absorption coefficients for the thick-layer components in the telescope. Again, the first column is a copy of the energy grid. The following columns are the mass absorption coefficients (in units of centimeters-squared per gram) for each thick material at that energy. Keywords in the header of the extension specify the material and density for each column.

PARAMETERS
telescop [string]
Mission name (value of the keyword TELESCOP to write in header keyword for CALDB.)

instrume [string]
Instrument name (value of the keyword INSTRUME to write in header keyword for CALDB.)

atomicfile [filename]
Name of the atomic data input file (FITS format with binary table extension.) If the extension is not specified in the parameter, the tool opens to the first binary table in the file. The five columns (and units) required are: 1. Z (Atomic Number), 2. Chemical symbol, 3. Name, 4.Weight (amu), 5.Density (grams per cubic centimeter) If the parameter 'atomicfile' is set to CALDB, the file is read from the calibration database.

atmscafile [filename]
Name of the atomic scattering input file (FITS format with binary table extension.) This file is used to calculate the optical constants (as a function of energy) of the telescope reflecting mirror surface materials. Different extensions of the file contain different compilations of data in the literature. The extension name can be specified in the extended file name format (name followed by extension name enclosed by []). If the extension is not specified in the parameter, the tool uses the value of the parameter 'atmsctng' as the extension name. This second parameter is also used if atmscafile is set to CALDB. The file contains the real and imaginary components of the

atomic scattering form factors as a function of energy in eV. Each extension of the file can contain a different compilation of form factors (the actual extension to be used is specified as part of the input file name string). The energy range of the form factors must cover the energy range for which the user is attempting to make a reflectivity/transmission file. The five columns (and units) required are: 1. Z (Atomic Number), 2.rowindex, 3. energy (eV), 4. f1real, 5. f2img.
Details of the compilations of atomic scattering factors in each extension, as well as references, can be found in the FITS file header. For the SXS+SXT-S and SXI+SXT-I, the name of the extension that should be used is HENKEMODSXT. For HXI1+HXT1 and HXI2+HXT2, the name of the extension that is used should be HenkeSskChantler.

atmsctng [string]
Name of the extension in the atomic scattering file (atmscafile) containing the atomic scattering data.

energyfile [filename]
The energy grid, either as a list of energies (in keV), or as a FITS file with the grid in a binary table extension named ENERGYGRID (in the latter case the units can be eV or keV). The column (and unit) required is ENERGY (eV or keV)

anglefile [filename]
Name of the angle grid input file. This must be a FITS file with a binary table extension named ANGLEGRID. The column (and unit) required is ANGLE (radian)

mirrorfile [filename]
Name of the mirror surface input file. This must be a FITS file with a binary table extension named SURFACE. This file contains the information about the layers (either single layer with only a substrate, or multi-layer with multiple layers of coatings above a substrate) of material on the surface of the mirror, separated into groups. The file should be sorted by group, then by layer. The eight columns (and units) required are:
group
firstshell (1st mirror shell for this group)
lastshell (last mirror shell for this group)
layer (coating layer number for this row and this group)
material (material constituting this layer)
density (density of this layer in grams per cubic centimeter)
thickness (thickness of this layer in Angstrom)
roughness (the roughness parameter for this layer in Angstrom)

outfile [filename]
Name of the output file for the reflectivity table.

outext [string]
Extension name for the output reflectivity table.

roughmodel [string]
Name of the model to calculate roughness (DW for Debye-Waller, or NC for Nevot-Croce, or NONE).

(roughkey = "ROUGHMOD") [string]
Keyword name to write in header to store the roughness model.

(validdate = 2999-01-01) [string]
The UTC date (in yyyy-mm-dd format) when this calibration data should first be used. This is the date as of which the xrtreftable output file is valid. This date is written to the output file in the keyword CVSD0001, required for any file entered into CALDB.

(validtime = 00:00:00) [string]
The UTC time (in hh:mm:ss format) on the day 'validdate' when this calibration data should first be used. This is the time as of which the xrtreftable output file is valid. This date is written to the output file in the keyword CVST0001, required for any file entered into CALDB.

desc [string]
CALDB description.

[ccldhm]

EXAMPLES
1. Run the tool, entering all the relevant information that is appropriate for the telescope HXT1.
xrtreftable atomicfile="atomicData_1997-11-01.fits[ATOMDATA]" atmscafile="atomic_scattering.fits[HenkeSskChantler]" \

energyfile="energy_grid_4to120keV.fits" anglefile="angle_grid_ah_hxt_reflect_20081103.fits" \
mirrorfile="ah_hx1_mirror_20131001v001.fits[SURFACE]" outfile="xrtreftable.fits" outext="HITOMI_HXT_FRONT" \
roughmodel="NC" telescop="HITOMI" instrume="HXT1" validdate="2014-01-01" validtime="00:00:00" \
desc="XRT foil reflectivity and transmission"

2. Run the tool with all the arguments in the user's ~/pfiles/xrtreftable.par file, without prompts to enter parameters.
 xrtreftable mode=h
