ASC Data Processing Tools for ACIS

Joel H. Kastner
AXAF Science Center
Science Data Systems

April 27, 1998

Contents

1 Tool Summaries **2**

1.1 Level 1 Tools 2

1.2 Level 2 Tools 2

2 Tool Descriptions **4**

2.1 Level 0.5 Tool 4

acis_collate_events 4

2.2 Level 1 Tools 6

acis_format_events 6

acis_correct_bias 12

acis_process_events 14

acis_grade_events 22

acis_calc_pi 26

acis_build_mask 28

acis_build_badpix 30

2.3 Level 2 Tools 32

acis_filter_events 32

acis_bin_events 34

acis_extract_spectrum 36

acis_extract_image 38

acis_extract_lightcurve 40

1 Tool Summaries

1.1 Level 1 Tools

acis_format_events Composite tool. For ACIS data obtained in faint event modes (TE faint, very faint, and faint with bias; CC faint), correct pixel pulse heights (ADU) for bias and overclock (tool: acis_correct_bias). For event data in either faint or graded modes, output ACIS exposure statistics table. Set various event status bits. Prepare event formats for further processing by acis_process_events.

acis_correct_bias For ACIS data obtained in faint event modes (TE faint, very faint, and faint with bias; CC faint), correct pixel pulse heights (ADU) for bias and overclock.

acis_process_events Composite tool. Perform calculations and corrections on ACIS event attributes. Correct event PHA for charge transfer inefficiency (optional); sum and grade events (tool: acis_grade_events); convert pulse height amplitudes (PHA) in ADU to pulse-invariant (PI) values in eV (tool: acis_calc_pi); convert chip coordinates to detector and sky coordinates. Set various event status bits.

acis_grade_events Grade ACIS events according to split geometry; sum PHA of pixels above split threshold, according to relevant grading system. Optionally, correct event PHA for charge transfer inefficiency (CTI) prior to grade (re-)determination and PHA summation.

acis_calc_pi Apply node-by-node gain coefficient table to convert pulse height amplitudes (PHA) in ADU to pulse-invariant (PI) values in eV.

acis_build_badpix Build bad pixel and column list file (single-CCD-specific), for later use in exposure map (and other processes TBD). List is created from archived and telemetered Bad Pixel (Column) Maps and any bias parity errors as reported in the science run Bias Error file for each CCD.

acis_build_mask Build spatial/PHA/event sampling mask for later use in exposure map (and other processes TBD). Mask is created from windows lists used by ACIS backend processors (BEPs) in accepting and rejecting events.

acis_summ_events Summarize results of a science run and/or OBI (i.e., Level 1 output) in format suitable for presentation to observer. Output includes ACIS setup summary, exposure statistics, event statistics, and per-CCD PHA grade splits, PHA histograms, etc. (TBR)

acis_format_hist For histogram mode data, add observation-specific keywords and make format compliant with XSPEC and ASCFIT requirements. (TBR)

acis_format_raw For data obtained in raw mode, find events and produce event list suitable for input to acis_format_events. (Mimics flight software processing steps for faint mode data.) (TBR)

1.2 Level 2 Tools

acis_filter_events Select ACIS events according to any or all available event attributes; in particular, select on event grade, position, energy, and status. Optionally, write output file of selected (filtered) events. Special-purpose wrapper for data_copy.

acis_bin_events Bin ACIS events into (1-D or 2-D) histogram. This is a general-purpose binning tool that serves as the core of special-purpose tools such as acis_extract_spectrum, yet is more specific in its function than data_bin_photons (around which it wraps).

acis_extract_spectrum Bin (appropriately filtered) ACIS events into PHA or PI histogram, and (optionally) calculate exposure time and effective area for selected parameter space.

acis_extract_image Bin (appropriately filtered) ACIS events into image, and (optionally) calculate exposure time and effective area for selected parameter space.

acis_extract_lightcurve Bin (appropriately filtered) ACIS events into light curve, and (optionally) calculate exposure time and effective area for selected parameter space.

acis_calc_splitratios Bin (appropriately filtered) ACIS events according to event grade, to find event split branching ratios.

acis_calc_pileupfrac Based on source count rate, estimate pileup fraction.

da_calc_hardness Calculate ACIS hardness ratios (definitions TBD) for a given source photon list or PHA (PI) histogram.

2 Tool Descriptions

2.1 Level 0.5 Tool

acis_collate_events

Description: For events obtained in "alternating" (aka "interleaved") exposure time TE (timed exposure) mode, collate Level 0 event list(s) and exposure records file(s) into 2 files each. Divisions of events and exposure records are made according to exposure time.

Parameters:

Inputs:

```
Level 0 parameter block file (*_pbk.fits)
Level 0 event list, 1 per active CCD
  (*_f_evt0.fits, where f is FEP number)
Level 0 exposure record file, 1 per active CCD
  (*_f_exr0.fits, where f is FEP number)
```

Outputs:

```
Level 0.5 event lists, 1 per exp time
  (*_A_evt0a.fits, *_B_evt0a.fits)
Level 0.5 exp record files, 1 per exp time
  (*_A_exr0a.fits, *_B_exr0a.fits)
```

Processing:

1. Open parameter block file for the science run. If value of `DTYCYCLE` keyword is nonzero (meaning that both primary *and* secondary exposure times are present in event list) then proceed with the following steps. If `DTYCYCLE` = 0 then no event or exposure collation is required, and steps below are not taken.

2. Open two output exposure records files, one for each exposure time (contained in parameter block keywords `EXPTIMEA` and `EXPTIMEB`). Set `CYCLE` keyword to reflect exposure time, e.g.:

   ```
   CYCLE   = 'P'       / events are from which exps? P[rimary]/S[econdary]/B[oth]
   ```

   where `CYCLE='P'` corresponds to `EXPTIMEA` and `CYCLE='S'` corresponds to `EXPTIMEB`.

3. Repeat preceding step, for the two output events files.

4. For each exposure record in each of the (up to six) exposure record files of the science run:

   (a) Use `EXPNO` to establish expected exposure time[1], based on values of parameter block keywords `DTYCYCLE`, `EXPTIMEA`, and `EXPTIMEB`:

   - If `EXPNO` = (1 + `DTYCYCLE`)$\times N$ ($N = 0, 1, 2, ...$) then exposure time is `EXPTIMEA`.
   - Otherwise, exposure time is `EXPTIMEB`.

[1] Need to establish whether first exposure has `EXPNO` value of 0 or 1.

(b) Check expected value of exposure time for exposure record against value of `EXPTIME` column of same record (should match, or there's a problem somewhere).

(c) Output exposure record to corresponding exposure records file.

5. For each event record in each of the (up to six) event files of the science run:

   (a) Use `EXPNO` to establish expected exposure time, based on values of parameter block keywords `DTYCYCLE`, `EXPTIMEA`, and `EXPTIMEB`:

   - If `EXPNO` = (1 + `DTYCYCLE`)$\times N$ ($N = 0, 1, 2,...$) then exposure time is `EXPTIMEA`.
   - Otherwise, exposure time associated with event is `EXPTIMEB`.

   (b) Output event record to corresponding events file.

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec20XX

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

2.2 Level 1 Tools

acis_format_events

Description: For ACIS data obtained in faint event modes (TE faint, very faint, and faint with bias; CC faint), correct pixel pulse heights (ADU) for bias and overclock (tool: acis_correct_bias). For event data in either faint or graded modes, output raw ACIS count rate and MTL tables. TBD: Merge (up to 6) CCD-specific event lists output by L0 into a single event list for all CCDs.

Parameters:

```
valid overclock (OC) ranges
various processing flags (see appendix to ‘‘Processing’’, below)
```

Inputs:

```
event data
  3x3 or 1x3 pixel ADU values (faint modes only)
  3x3 bias values (TE faint w/ bias only)
  chipx,y of central pixel (CC: chipx only)
  exposure number
  time
  CCD ID
  PHA (graded modes only)
bad pixel map
bias map (N/A for TE faint w/ bias mode)
exposure record data
  exposure number
  OC values
  # pixels above threshold
  # events telemetered
```

Outputs:

```
bias/OC-corrected event data
updated bias map (TE faint w/ bias only)
updated MTL file
count rate table
OC table
```

Processing:

1. Determine ACIS mode (one of TE faint, TE very faint, TE faint with bias, or CC faint)
2. Read event, bias (if nec.), bad pixel, and exposure data.
3. For each active CCD, loop over events:
   (a) Correct for bias and overclock (tool: acis_correct_bias).
   (b) Update event/pixel count table using data (events telemetered and pixels above threshold) from exposure record.

4. TBD: Merge (up to 6) time-ordered CCD-specific event lists output by L0 into a single, time-ordered event list for all CCDs.

Appendix A: Design Description (W. McLaughlin, 3/25/97)

```
main
  |_ evtfntformat
       |_ parse_acis_evt_columns
       |_ determine_acis_mode  (A)
       |_ load_bias_image (A)
       |_ open_exposure_file  (A)
       |_ evtfnt_dependency_check
       |_ keep_input_file_axes  (A)
       |_ open_mtl_file  (M)
       |_ open_cntrt_file  (C)
       |_ load_event_data  (A)
       |_ process_mtl_update  (M)
       |_ bias_correct
       |    |_set_bias_map_status_bits
       |
       |_ faint_bias_extraction
       |    |_ check_file_existance  (A)
       |
       |_ load_cntrt_buffer  (C)
       |_ write_cntrt_update  (C)
       |_ load_exposure_column  (A)
       |_ apply_oc_corrections
       |_ write_acis_events  (A)
       |_ evtfnt_log_errors


EVTFNDFORMAT()
BEGIN
   get parameters
   set up stacks
   IF (size of input and bias stacks differ) THEN
      WHILE {input stack not null} LOOP
         IF {no error opening input file} THEN
            set up input file column masks
            IF {no error opening bias file}  THEN
               allocate memory for bias table
               load bias table
               IF {output file not set up} THEN
                  IF {no errors opening output file} THEN
                     setup output file
                  ELSE
                     add err mask to status flag (out open err)
                  ENDIF
               ENDIF
               close bias file
```

```
            ELSE
               add err mask to status flag (bias open err)
            ENDIF
         ELSE
            add err mask to status flag (input file open err)
         ENDIF

         IF {no errors encountered} THEN
            WHILE { not all events processed} LOOP
               load event
               bias adjust event
               overclock correct event
               write out event
            ENDWHILE
         ENDIF
         close input file
      ENDWHILE
      close output file
   ELSE
      add err mask to status flag (diff size stacks)
   ENDIF
END.
```

Appendix B: Description of Parameters (W. McLaughlin, 6/27/97)

```
The parameters listed below provide evtfntformat() information on how to
execute.

   infile     - the input qpoe file(s). This value may either be a
                a qpoe filename or the name of a stackfile containing one or
                more qpoe file names. If the later is the case, the value
                must be proceded by a '@' so that the routine knows that it must
                expand the stack. Pathnames may included in the input value.

   outfile    - the output qpoe file. The name may be either a name or
                a path and name.

   badpix     - This value specifies the bad pixel map file to use. it may
                either be a file name or "NONE". If "NONE" is used, bad pixel
                map status bits will not be set. If a file is used, the value
                may include the file's path and should be a fits file.

   biasfile   - This parameter specifies the name of the bias map to use to
                apply bias corrections to input events. For faint with bias
                data, if this value is set to "NONE", the bias columns of the
                input event file are used. If it is set to a file name, the
                file is used for bias corrections. Provided bias files are
                expected to be fits image files. A separate bias file should
```

be provided for each active ccd if bias map corrections are to be applied.

expname - This parameter specifies the exposure file to use when applying overclock corrections, writing out the count rate and mission time line files. the value of this parameter should be either a file name or "NONE"

outbias - This parameter specifies the stem of the output bias maps generated when a faint with bias event file is processed by evtfntformat. A separate bias map for each active ccd (which has at least one event hit) is generated. the created file name consists of the specified stem + the ccd number + .fits

cntrt_file - This parameter provides the name of the count rate file which is generated by this routine.

mtl_file - This parameter provides the name of the mission time line file which is generated by this routine.

logfile - This value specifies the name of the ascii text file which evtacisdet will generate if the debug parameter (see below) is set to a value other than 0. If the value is set to "stdout", the output will be redirected to standard output (typically the screen)

eventdef - This field allows the user to specify the contents of the input file. The value may either be set to a desired list of columns and data types by the user, or a redirect to predefined event definitions may be utilized via the redirect command.

datatype - This parameter tells the tool which way the pixels in an event island are ordered. This is important since FLIGHT and SAC/ACIS-2C have different internal representations.

bias_correct - This parameter serves as a flag to instruct evtfntformat as to whether or not it should perform bias corrections.

oc_correct - This flag allows the user to specify whether or not overclock corrections are applied to the events.

min_init_oc - This value identifies the minimum acceptable value for an initial overclock.

max_init_oc - This value identifies the maximum accpetable value for an initial oveclock.

min_dlta_oc - the minimum acceptable delta overclock value is specified by this value. If an exposure file entry's delta overclock values fall below this value a status bit is set in the respective

```
              event.

max_dlta_oc - the maximum acceptable delta overclock value is specified by
              this value. If an exposure file entry's delta overclock values
              are above this value a status bit is set in the respective
              event.

qp_internals - (n/a)- This boolean parameter instructs evtacisdet whether
             or not to use the the page and bucket length values specified
             in the input file or to use the default values.

qp_pagesize - (n/a) - allows the user to specify the qpoe page size

qp_bucketlen - (n/a)- allows the user to specify the qpoe bucket length

debug      - allows the user to request a varying level of textual
             output based upon the program execution. Levels are from 0-5
             with 0 representing no information and 5 representing as much
             detail as possible.

tempbias   - This flag is a temporary work around which causes evtfntformat
             to perform a simple bias correction algorithm if bias maps are
             unavailable.

clobber    - This parameter instructs evtfntformat to remove an already
             existing file so that a new bias map file may be created with
             the same name- in effect 'overwriting' or 'clobbering' the
             previous output file.


telev1     - This event definition specifies the default output columns that
             will be written to the output file if the eventdef variable is
             redirected here.

vflev1     - This event definition specifies the default output columns that
             will be written to the output file if the eventdef variable is
             redirected here. It is primarilly intended from use with very
             faint mode data (ie. 5x5 mode data).

cclev1     - This event definition specifies the default output columns tha
             will be written to the output file if the eventdef variable is
             redirected here. It is primarily intended for coninous clocking
             mode data.
```

Release: 4

Group: DA

Analysis Domain: Event

DS Tool Class: 3

DS Tool Category: Correction

Spec Name: acis/spec/spec78

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC

acis_correct_bias

Description: For ACIS data obtained in faint event modes (TE faint, very faint, and faint with bias; CC faint), correct pixel pulse heights (ADU) for bias and overclock.

Parameters:

```
valid overclock (OC) ranges
```

Inputs:

```
event data
  3x3 or 1x3 pixel ADU values (faint modes only)
  3x3 bias values (TE faint w/ bias only)
  chipx,y of central pixel (CC: chipx only)
  exposure number
  time
  CCD ID
  PHA (graded modes only)
bad pixel map
bias map (N/A for TE faint w/ bias mode)
exposure record data
  exposure number
  OC values
  # pixels above threshold
  # events telemetered
```

Outputs:

```
bias/OC-corrected event data
updated bias map (TE faint w/ bias only)
OC table
```

Processing:

1. Make correspondence between event pixel ADU values and appropriate bias values, and correct event pixel ADU for bias (i.e. subtract bias ADU from event ADU on pixel-by-pixel basis). `[Faint with bias only: update ‘‘real-time’’ bias map.]`
2. Correct event ADU for overclock (OC). OC is a single, per-exposure value for each CCD node, consisting of initial OC (OC value at start of science run) + delta OC (exposure-by-exposure change in OC); both initial and delta OC's are found in the exposure record (*_exr.fits) files output by L0. Update overclock table (write initial + delta OC).

Release: 4

Group: DA

Analysis Domain: Event

DS Tool Class: 3

DS Tool Category: Correction

Spec Name: acis/spec/spec22

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC

acis_process_events

Description: Perform calculations and corrections on ACIS event attributes. Sum and grade events (see acis_grade_events); convert chip coordinates to detector and sky coordinates; convert pulse height amplitudes (PHA) in ADU to pulse-invariant (PI) values in eV. (Optionally, also correct event PHA for charge transfer inefficiency [CTI].)

Parameters:

```
grading scheme (flight bitmap to ASCA/ACIS)
split thresholds
various processing flags (see appendix to ‘‘Processing’’, below)
```

Inputs:

```
event list (output of acis_format_events)
ACIS detector geometry
dither (aspect) history
gain table or response matrix
```

Outputs:

```
Updated event list (w/ grades, det/tdet/world coords, PI)
Event Log
```

Processing:

1. Grade event and sum event pixel PHA (tool: acis_grade_events).
2. Correct PHA for charge transfer inefficiency (under study); calculate PI based on CCD gain (tool: acis_calc_pi).
3. Calculate tiled detector coordinates: receive the graded and summed event list, with event location in chip coordinates. Apply the chip to tiled detector coordinate transformation for each event and write the detector coordinates to the event list.
4. Calculate world (celestial) coordinates: transform the event local maximum position from detector coordinates to world coordinates.
5. Set event status bits (see Appendix below; most status bits are set in the preprocessor tool acis_format_events).
6. Write updated event record to event list.

All necessary processing information (including debugging if desired) is recorded to the events log.

Appendix A: Design Description (W. McLaughlin, 3/24/97)

```
A calling tree of the evtacisdet tool is listed below... The detailed design
of each of the routines listed in the tree is also provided.
```

```
Note: several of the routines listed below are actually used by several level
      1 acis tools and have therefore been moved into a separate library
      (acisio lib). They are designated by a (A) after the function name.

main
  |_ evtacisdet
       |_ read_instrume_params (A)
       |_ parse_coord_range
       |_ map_start_column
       |_ verify_scheme_request
       |_ load_short_key_value (A)
       |_ set_grating_type
       |_ load_double_key_value (A)
       |_ parse_acis_evt_columns (A)
       |_ determine_acis_mode (A)
       |_ determine_island_size (A)
       |_ dependency_check_acis
       |_ set_up_mirror
       |_ setup_output_axes
       |_ write_instrume_params (A)
       |_ setup_focal_length (A)
       |_ get_predicted_beam_position
       |_ load_event_data (A)
       |_ dither_update
       |    |_ open_dither_file
       |    |    |_ map_table_column
       |    |    |_ dither_file_dependencies
       |    |
       |    |_ load_dither_entry
       |    |_ close_dither_file
       |
       |_ process_grades_acis
       |    |_ sum_grade_event
       |    |_ find_acis_grade
       |    |_ find_asca_grade
       |
       |_ calculate_coords_acis
       |    |_ calc_chip_coords
       |    |    |_ calculate_centroid
       |    |
       |    |_ calc_sky_coords
       |    |_ calc_tan_coords
       |    |_ calc_det_coords
       |
       |_ write_acis_events (A)
       |_ log_warning_stats
       |_ make_wcs_updates
       |_ log_error_stats
```

Appendix B: Description of Parameters (W. McLaughlin, 5/22/97)

```
The specific contents of the output file depend upon a combination of the input
file and the control options selected. Listed below is a brief summary of the
various control options (parameters) available and which of the two main
evtacisdet functions the option applies to (conv)ersion, (grad)ing, or (both),
These parameters are contained in the evtacisdet.par parameter file and can be
set in the parameter file or as command line arguments.

   poefile    - (both) - the input qpoe file(s). This value may either be a
                a qpoe filename or the name of a stackfile containing one or
                more qpoe file names. If the later is the case, the value
                must be proceded by a '@' so that the routine knows that it mus\
t
                expand the stack. Pathnames may included in the input value.

   outfile    - (both) - the output qpoe file. The name may be either a name or
                a path and name.

   dither     - (conv) - the input dither file or stack of files. This informs
                evtacisdet what dither file to use to perform dither correction\
s
                on the fpc coordinate data. If a value of 'NONE' is specified
                dither corrections will not be performed.

   logfile    - (both) - This value specifies the name of the ascii text file
                which evtacisdet will generate if the debug parameter (see
                below) is set to a value other than 0. If the value is set to
                "stdout", the output will be redirected to standard output
                (typically the screen)

   eventdef   - (both) - This field allows the user to specify the contents of
                the input file. The value may either be set to a desired list
                of columns and data types by the user, or a redirect to
                predefined event definitions may be utilized via the redirect
                command.

   datatype   - (both) - This parameter tells the tool which way the pixels in
                an event island are ordered. This is important since FLIGHT
                and SAC/ACIS-2C have different internal representations.

   doevtgrade - (grad) - option to calculate and output the flight grade and
                and pulse height sum of events in the input qpoe file.

   scheme     - (grad) - option to select a grading scheme for events in the
                input qpoe file. Valid grading schemes are ASCA, ACIS, or NONE.

   spthresh   - (both) - the user defined split threshold value used in
                determining centroid calculations as well as flight grade and
                pulse height sum values.
```

time_offset- (conv) - This value allows the user to specify a synchronizatio\
n
value to use in matching the dither file records to the event file records. It may be a positive or negative value that will be added to the event time used in determining the correct dither interval. It's units are in seconds.

docentroid - (conv) - option to adjust the location of the local maxima base\
d
upon weighted values of surrounding pixel pulse heights above a threshold value.

tstart - (both) - This value provides the default time for events from the input data file. If the input data does not have a time field, this value will be used. The value is also used to ensure that when multiple input files are 'stacked', that they are ordered chronologically without any overlap.

tstop - (both) - This value indicates the tail end of the range of events in the input file. If its value is equivalent to that of the input minimum 'tstart', an epsilon value will be added. It is used to ensure that 'stacked' input files are ordered chronologically without any overlap.

qp_internals - (n/a)- This boolean parameter instructs evtacisdet whether or not to use the the page and bucket length values specified in the input file or to use the default values.

qp_pagesize - (n/a) - allows the user to specify the qpoe page size

qp_bucketlen - (n/a)- allows the user to specify the qpoe bucket length

debug - (both) - allows the user to request a varying level of textual output based upon the program execution. Levels are from 0-5 with 0 representing no information and 5 representing as much detail as possible.

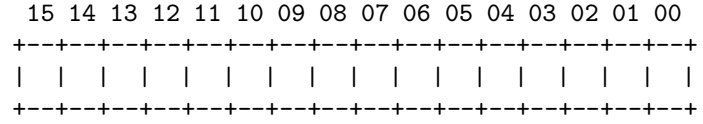
start - (conv) - This parameter specifies the starting point of the coordinate transformations. If no coordinate transformations are to be applied the value should be set to 'chip' (which is the default value). This value should only be changed to some other value if the transformations are to be run from a point other than chip coords (ie. re-running evtacisdet on a data set that has already been processed and contains tdet coords but does not contain chip coords).

stop - (conv) - This parameter allows the user to specify the final coordinate transformation that will be performed on the data. If the value is set to 'none' then no transformations will take place. Otherwise all coordinate transformations from chip to

```
              the specified value will take place (ie. if stop is set to
              sky, then det, tdet, tan, and sky coords will be calculated).

instrume    - (conv) - The instrume parameter allow the user to specify the
              type of instrument to use when processing the data (ie. acis
              or acis-2c). System parameters and staging/mirror alignment
              correction values will be loaded from a parameter file with
              the name specified in this parameter field.

telescop    - (conv) - This value allows the user to specify the name of the
              telescope which will be used to set up the focal length. If the
              value specified is "" then evtacisdet will default to using the
              telescop keyword from the header file to determine the length
              to use as the focal length. Acceptable values include: "HRMA",
              "XRCF-TMA", and "XRCF-HRMA".

random      - (conv) - When set, this flag will instruct the pixel library to
              randomize the value of sky coords before writing them out. This
              is designed to compensate for the anti-aliasing striping that
              will occur when the calculated double precision values are
              caste to float.

stdlev1     - This event definition specifies the default output columns that
              will be written to the output file if the eventdef variable is
              redirected here.

5x5lev1     - This event definition specifies the default output columns that
              will be written to the output file if the eventdef variable is
              redirected here. It is primarilly intended from use with very
              faint mode data (ie. 5x5 mode data).

cclev1      - This event definition specifies the default output columns tha
              will be written to the output file if the eventdef variable is
              redirected here. It is primarily intended for coninous clocking
              mode data.
```

Appendix C: Event Status Bits (summarized by W. McLaughlin, 3/25/97)

NOTE: Bits 01,02,03 are set in evtacisdet; bits 04,06,07,08,09,10 and 12,13,14 are set in evtformat. Bits 00,05,11 are presently (8/97) not implemented.



STATUS	MASK VALUE
L1_STS_GOOD_EVENT	0x0000
L1_STS_BAD_POS	0x0001
L1_STS_LM_BAD_VAL	0x0002
L1_STS_PIX_RANGE	0x0004
L1_STS_BAD_SUM	0x0008
L1_STS_LM_BAD_PIX	0x0010
L1_STS_ADJ_BAD_PIX	0x0020
L1_STS_BIAS_BAD_PIX	0x0040
L1_STS_BIAS_MISSING	0x0080
L1_STS_BIAS_PARITY	0x0100
L1_STS_OC_MISSING	0x0200
L1_STS_OC_RANGE	0x0400
L1_STS_CM_LOW	0x0800
L1_STS_CM_1_MISSING	0x1000
L1_STS_CM_2_MISSING	0x2000
L1_STS_CM_3_MISSING	0x3000
L1_STS_CM_4_MISSING	0x4000

EVENT POSITION

- 00 - This bit is set to 1 if an event's chip coordinates fall outside of the valid region. This may be due to windowing, active-inactive ccds, etc...

PULSE HEIGHTS

- 01 - This bit is used to flag events in which the center pixel of the 3x3 event island does not contain the local maxima (highest pha value in the island).

- 02 - This bit is set to 1 for any event which has one or more of its individual island pixels > 4095.

- 03 - This bit is set to 1 to announce the detection of an overflow condition in the summing of event island columns. Since 'pha' (sum of event island pulse heights) have a range from 0-36855 and the pha category can only handle short integers (numbers upto 32767), this bit will flag events which exceed the limit. In addition, the 'pha' sum of such an event will be set to the default value of 32767.

BAD PIXELS

- 04 - A value of 1 in this bit means that the event's local maxima falls on a pixel identified in bad pixel map as being 'bad'.

- 05 - This bit is used to indicate that one or more of an event's edge or corner pixels falls on a pixel identified within the bad pixel map.

BAD BIAS

- 06 - This pixel flags events which have bad bias map values. The flag is set when bias values are set to 4095 by the onboard software.

```
   - 07 - This bit position indicates missing or unknown bias values for an
          event pixel. This bit is set due to situations such as telemetry
          dropout.

   - 08 - Parity errors are recorded for events by setting this bit to 1. The
          bit is set when the onboard software returns a bias value of 4094.

BAD OVERCLOCK VALUES
   - 09 - This bit is set for events for which overclock information is missing
          or does not exist.

   - 10 - Setting this status bit to 1 indicates that the corresponding event's
          overclock values fall outside of a specified range (nominal case
          0>= oc_val >= 500)

BAD CORNER MEAN (GRADED MODE ONLY)
   - 11 - corner mean below -4095
          This bit is used to flag events where the onboard software has
          set the corner mean value to -4096 to indicate an extremely low corne\
r
          mean value (see ECO 36-946).

   - 12/13 - used to indicate the number of corners missing from the
          corner mean (ie bit pattern 01=1 missing, 10=2 missing, and 11=3
          missing corners). These bits will be set based on the results of
          checking the bad pixel and bad bias values of the corners surrounding
          the center position of graded mode data. These bits tell the number
          of corners missing but does not identify which corners are missing.
          These bits should be mutually exclusive from bit 14 (see below).

   - 14 - corner mean missing (ie. no valid corner pixels)
          This bit is set to 1 if the corner mean value telemetered from the
          onboard software is set to 4095 (see ECO 36-946) and represents a
          graded mode event which has no valid corner pixels. Whether or not to
          set this if the level 1 software receives an event with 4 bad corners
          that wasn't flagged with a value of 4095 by the onboard software, is
          polemical.


Notes: The status word and bit masks identified here are specific to front-end
       level 1 processing. Back-end level 1 status bits (TBD) will be logically
       grouped- either as an independant status word or as the high end bits of
       a 32 bit long.

       Bit position 15 is currently unused.
```

Release: 4

Group: DA

Analysis Domain: Event

DS Tool Class: 3

DS Tool Category: Correction

Spec Name: acis/spec/spec24

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC

acis_grade_events

Description: Grade ACIS events according to split geometry; sum PHA of pixels above split threshold, according to relevant grading system. Optionally, correct event PHA for charge transfer inefficiency (CTI) prior to grade (re-)determination and PHA summation.

Applicable ICDs:

```
ARD (http://space.mit.edu/ASC/docs)
ACIS Level 1 (http://space.mit.edu/ASC/docs)
```

Parameters:

```
split thresholds
```

Inputs:

```
event pixels
grades table (flight bitmap to ASCA/ACIS) (ARD)
CTI coefficient table (ARD)
```

Outputs:

```
flight grade (0-255)
ASCA grade (0-7) --or-- ACIS grade (TBD)
```

Processing:

1. From `DATAMODE` keyword, establish readout mode (TE or CC) and telemetry packing mode (for TE, could be faint, faint with bias, very faint, or graded; for CC, could be faint or graded).
2. OPTIONAL: If events obtained in timed exposure (continuous clocking) *faint* modes, correct all 9 (3) event PHA values for charge transfer inefficiency (CTI) using node-by-node lookup table of CTI coefficients in acis_CTI.fits (described in ACIS Analysis Reference Data ICD, see `http://space.mit.edu/ASC/docs`). The corrected values are then used to determine the event grade (see below).

   If events obtained in timed exposure or continuous clocking *graded* mode, correct total event PHA, via same algorithm.

   (Whether to replace "raw" PHA values with CTI-corrected PHA values in event list is TBD.)

   (a) Determine CCD node from event `CHIPX`, based on the following table:

Node	min(`CHIPX`)	max(`CHIPX`)
0	0	255
1	256	511
2	512	767
3	768	1023

   (b) For each pixel of *faint* mode event (or for central pixel of *graded* mode event) determine pixel positions in `READ` coord system, as a function of CCD node (note: following table only applies to standard [ABCD] node readout):

Node	READX	READY	min(READX)
0	CHIPX	CHIPY	0
1	767− CHIPX	CHIPY	256
2	CHIPX	CHIPY	512
3	1791− CHIPX	CHIPY	768

(c) The following two steps are performed iteratively until corrected PHA converges and/or max. (nominally ~ 10) iterations are reached:

(d) Determine parallel and serial (alternatively, X and Y, or CCD column and row) charge loss from CTI coefficients as follows:

$$C_X = P_b(n) \times (\mathrm{PHA}^{P_a(n)})$$

$$C_Y = S_b(n) \times (\mathrm{PHA}^{S_a(n)})$$

where C_X and C_Y are the parallel and serial charge loss; P_a, P_b and S_a, S_b are the parallel and serial CTI coefficients, respectively; n is the node; and PHA is the latest (corrected) event PHA (equal to PHA′, below). Initially, set PHA = PHA_0, where PHA_0 is the raw (original, uncorrected) event PHA.

(e) Calculate CTI-corrected PHA (PHA′) as

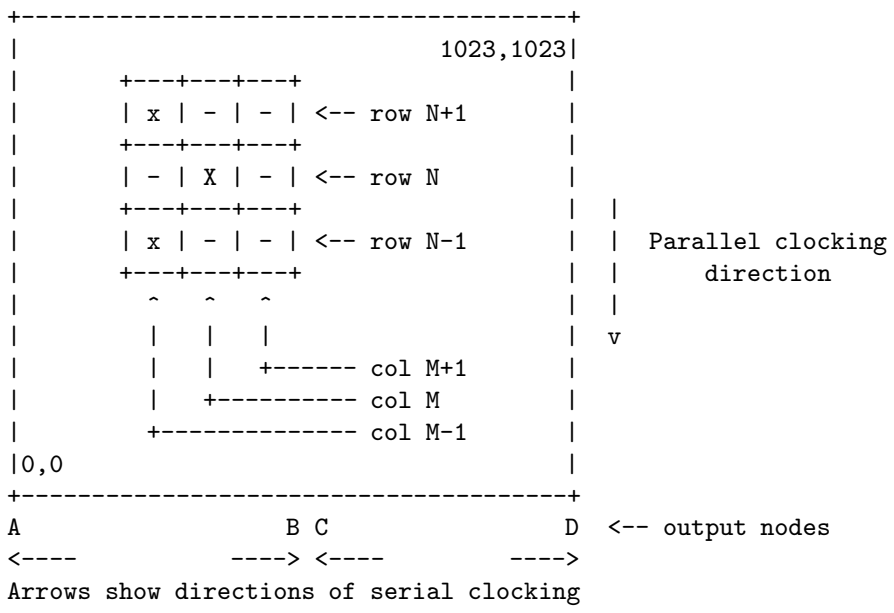
$$\mathrm{PHA}' = 0.5 + \mathrm{PHA}_0 \times (1.0 - \mathrm{READY} \times C_Y - (\mathrm{READX} - \min(\mathrm{READX}) \times C_X))^{-1}$$

where the values of min(READX) depend on node and are obtained from the previous table.

3. For each event obtained in timed exposure (continuous clocking) *faint* modes, grade events and sum pulse heights, as follows:

(a) The hexidecimal flight grade (or a value from 0-255 in base 10, for TE mode) is calculated from a pixel bitmap constructed the 8 (2) corrected PH values of the pixels surrounding the local maximum, based on the bitmap definitions for TE (CC) mode as described in the ACIS IP&CL software structure definitions (§5.7 of Rev. G, 1/97). The following schematic view is from a Peter Ford email of 6/16/97:

Here's the way it is for the SRS example, grade 33 (0x21). We're looking at the CCD from the HRMA side and the output nodes are at the bottom of the figure. The event is reported at ccdRow=N, ccdColumn=M.



If the pixel PHA is greater or equal to the split threshold value (note: split threshold is CCD dependent, with different values for FI and BI devices), then the corresponding bit is marked, otherwise it stays 0; hence, for the above example, `001 00 001 = 0x021 (hex) = 33 (base 10).` For continuous clocking (CC) data, the bitmap just consists of bits in the right and left pixels next to the local maximum (hence has a value from 0-4 in base 10).

(b) Calculate the total event amplitude (PHA): Sum the corrected pulse height values of the central pixel and all surrounding pixels that are deemed part of the event, according to the grading system in use. In the ASCA system, corner pixel PHAs are generally excluded, whereas in the ACIS system, corner pixel PHAs will be included (to be consistent with ACIS flight software definitions).

4. Based on the flight grade just calculated for faint modes, or as telemetered in the event record in the case of graded modes, assign an ASCA-style quality (grade) in either the ASCA or (TBD) ACIS systems. (For CC data, we somewhat arbitrarily define grades as follows, if the desired grading system is 'ASCA': 0 for single pixel events, 3, 4 for left and right splits, respectively, and 7 for 3-pixel splits.) See Analysis Reference Data ICD for definition of file that maps "flight" grade to "quality" (ASCA or ACIS) grade.

Release: 4

Group: DA

Analysis Domain: Event

DS Tool Class: 3

DS Tool Category: Correction

Spec Name: acis/spec/spec23

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC

acis_calc_pi

Description: Convert pulse height amplitudes (PHA) in ADU to pulse-invariant (PI) values in eV.

Applicable ICDs:

```
ARD (http://space.mit.edu/ASC/docs)
ACIS Level 1 (http://space.mit.edu/ASC/docs)
```

Parameters:

Inputs:

```
event list
gain coefficient table (ARD)
```

Outputs:

```
Updated event list (w/ PI column)
```

Processing: For each event in an event list:

1. Verify that `PHA` column is present in event list (will be true of data obtained in *graded* mode as well as *faint* mode data that has been processed by acis_grade_events or acis_process_events).
2. Determine CCD node from event `CHIPX`, based on the following table:

Node	min(`CHIPX`)	max(`CHIPX`)
0	0	255
1	256	511
2	512	767
3	768	1023

3. Using node-by-node lookup table of gain coefficients in acis_gain.fits (described in Analysis Reference Data ICD; see `http://space.mit.edu/ASC/docs`), convert event PHA to PI:

$$\mathrm{PI} = a_0 + a_1 \times \mathrm{PHA} + a_2 \times \mathrm{PHA}^2 + a_3 \times \mathrm{PHA}^3$$

where a_i are the gain coefficients called, respectively, `OFFSET` (a_0), `GAIN_ORD1` (a_1), `GAIN_ORD2` (a_2), and `GAIN_ORD3` (a_3) in acis_gain.fits.

Release: 4

Group: DA

Analysis Domain: Event

DS Tool Class: 3

DS Tool Category: Correction

Spec Name: acis/spec/spec2023

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC

acis_build_mask

Description: Build spatial/PHA/event sampling mask for later use in exposure map (and other processes TBD). Mask is created from windows lists used by ACIS backend processors (BEPs) in accepting and rejecting events. Each CCD can have a set of up to 6 BEP spatial acceptance windows; furthermore, each window has an associated PHA acceptance range and event sampling interval (the "sample cycle").

Parameters:

Inputs:

```
from parameter block file header (*_ccpbk.fits or *_tepbk.fits):
  subarray definition, CCD coords
from 1-D (CC) or 2-D (TE) window list file (*_win1.fits or *_win2.fits):
  backend processor window table (one per active CCD):
    CCD_ID
    lower left corner position, CCD coords (CC mode: first CCD column)
    window X,Y dimensions (CC mode: X only)
    accepted event PHA range
    event sample cycle, N (i.e., accept every Nth event)
```

Outputs:

```
subarray definition, CHIP coords (header keywords)
window list table
  CCD_ID
  window precedence
  spatial description, CHIP coords
  PHA range
  sample cycle
```

Processing:

1. Read subarray start row `STARTROW` and subarray rows `ROWCNT` from parameter block file header. Calc. keywords `FIRSTROW` and `NROWS` as follows:

   ```
   FIRSTROW = STARTROW + 1
   NROWS = STARTROW + ROWCNT + 1
   ```

   and write to header of *_msk FITS file.

2. Read in window list from each 2D (CC: 1D) window definition file included in science run telemetry. [If no window files are included in telemetry for a given ACIS science run, see below.]

3. For each L0 window table entry, calculate and store the following in one row of the output mask table:

   ```
   WINDOW (running value from 1 up to, but not exceeding, 6 for each CCD)
   LL_CHIPX = LL_CCDX + 1
   LL_CHIPY = LL_CCDY + 1
   UR_CHIPX = LL_CCDX + CCDCOL + 1
   UR_CHIPY = LL_CCDY + CCDROW + 1
   ```

The entry for `WINDOW` indicates the order in which windows were applied in accepting or rejecting events. Thus the window ordering scheme of the ACIS flight software is made explicit here. Note that the first window listed for a given CCD is assigned `WINDOW` = 1 and was the first one applied when accepting/rejecting events; the 2nd window listed for a given CCD is assigned `WINDOW` = 2 and was the 2nd one applied when accepting/rejecting events; etc.

The L1 table entries for `CCD_ID, SAMP_CYC, PHAMIN, PHARANGE` retain their L0 values.

4. Case of no window files output by telemetry: define one window (row in table) per CCD, as follows:

```
CCD_ID = [0 to 9]
WINDOW = 0
LL_CHIPX = 1
LL_CHIPY = 1
UR_CHIPX = 1024
UR_CHIPY = 1024
SAMP_CYC = 1
PHAMIN   = 0
PHARANGE = 65535
```

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec31

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

acis_build_badpix

Description: Build bad pixel and column list file (single-CCD-specific), for later use in exposure map (and other processes TBD). List is created from archived Bad Pixel Map and any bias parity errors as reported in the science run Bias Error file for each CCD. As an (optional) second step, a second list is also compiled from the Bias Map for the same science run or, in the absence of a telemetered bias, from the most recent Bias Map obtained in the same ACIS data-taking mode as the science run in question. [For faint with bias mode, this second bad pixel list is obtained from the event-by-event bias values.] The two lists are then compared and combined, using a bitmap column to indicate the origin of each bad pixel (i.e., archived bad pixel list, bias map, and/or bias parity error list).

Parameters:

Inputs:

```
from archived Bap Pixel List:
  bad pixel list
  bad column list
from Bias Map image file (*_bias.fits):
  bad pixel map (i.e. pixels assigned value 4095)
from Bias Error table file (*_berr.fits):
  bias parity error list
```

Outputs:

```
bad pixel list (table):
  time
  CHIPX,CHIPY
  origin (bitmap)
bad column list (table):
  CHIPX
```

Processing:

1. Read and injest current archived bad pixel & column list for relevant CCD. Bad pixels are stored in 1st table extension, bad columns in 2nd table extension (TBR). This list is compiled in `CHIPX/Y` coordinate system (TBR), so no coordinate transformations need be applied.
2. Read and injest lists from Bias Error file, if one exists for the relevant CCD.
3. Optional step (i.e., if bias data available and appropriate processing flag set):
   (a) Faint or graded mode data: From input Bias Map, form list consisting of positions (in CHIP coord system) of pixels assigned values 4095. In the ideal case, this list should be identical to the "permanent" bad pixel list read in from archive.
   (b) Faint with bias mode data: From bias values accompanying event data, form list consisting of positions (in CHIP coord system) of pixels assigned values 4095. In the ideal case, this list should be a subset of the "permanent" bad pixel list read in from archive.
4. Compare (up to 3) list(s), merge (if nec.), and set `ORIGIN` bits:

```
Bit 0: from archived list
Bit 1: from bias error file
Bit 2: from bias data
```

5. Write merged bad pixel list to 1st extension of output Bad Pixel table file:

   ```
   TIME
   CHIPX
   CHIPY
   ORIGIN
   ```

   The value in the `TIME` column depends on the origin of the pixel. For pixels with the 0th `ORIGIN` bit set it is the archived Bad Pixel List file creation time. For pixels with the 1st `ORIGIN` bit set it is the time derived from the corresponding Bias Error table file entry. For pixels with the 2nd `ORIGIN` bit set it is `TSTART` of the OBI. For more than one of the above, the largest (latest) value of `TIME` takes precedence.

6. Write bad column list (consisting of one column, `CHIPX`) to 2nd extention of output Bad Pixel table file.

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2032.tex

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

2.3 Level 2 Tools

acis_filter_events

Description: Select ACIS events according to any or all available event attributes; in particular, select on event grade, position, energy, and status. Optionally, write output file of selected (filtered) events. Special-purpose wrapper for `data_copy`.

Parameters:

```
Filter criteria; admissible values for:
  TIME
  X,Y (e.g. regions file)
    CHIP, TDET, DET or sky (RA,dec) coord systems
  PHA
  PI
  GRADE
  CCDID
  STATUS
```

Inputs:

```
``Raw'' event list
```

Outputs:

```
Filtered event list
Filter criteria (as header keywords)
```

Processing:

1. Open parameter file and determine filter criteria.
2. Open input event file.
3. Apply filter criteria to select events.
4. Optionally, write output file of selected (filtered) events. Include header keywords specifying filters applied.

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2035

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

acis_bin_events

Description: Bin ACIS events into (1-D or 2-D) histogram. This is a general-purpose binning tool that serves as the core of special-purpose tools such as `acis_extract_spectrum`, yet is more specific in its function than `data_bin_photons` (around which it wraps).

Parameters:

```
Event attribute over which to bin:
  TIME
  X,Y (e.g. regions file)
    CHIP, TDET, DET or sky (RA,dec) coord systems
  PHA
  PI
  GRADE
Bin width
Bin range
```

Inputs:

```
event list
```

Outputs:

```
histogram (or image, if selected event attribute is X,Y)
```

Processing:

1. Read parameter file and establish event attribute over which to bin (and, implicitly, dimensionality of resulting histogram). Evaluate binning parameters (bin width, bin range).
2. Open and read events file. (Evaluate keywords for presence of applied filter(s)).
3. Initialize histogram, based on event attrribute and bin parameters.
4. Loop over events, incrementing histogram location indexed by the selected event attribute.
5. Optional (if event filters present): Read exposure map, and bin in like manner to derive exposure time and effective area.
6. Write histogram file.

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2036

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

acis_extract_spectrum

Description: Bin (appropriately filtered) ACIS events into PHA or PI histogram, and (optionally) calculate exposure time and effective area for selected parameter space. Wrapper around `acis_filter_events` and `acis_bin_events`.

Parameters:

```
Histogram type (PHA or PI)
Filter criteria for event attributes:
  TIME
  X,Y (e.g. regions file)
    CHIP, TDET, DET or sky (RA,dec) coord systems
  GRADE
  STATUS
  CCDID
Bin width
Bin range
```

Inputs:

```
event list
exposure map
```

Outputs:

```
histogram (PHA or PI)
region exposure time
region effective area
```

Processing:

1. Read parameter file and establish event attribute over which to bin (PHA or PI). Evaluate filter criteria and binning parameters (bin width, bin range).
2. Open and read events file.
3. Initialize histogram, based on event attribute and bin parameters.
4. Loop over events:
   (a) Apply filter criteria to select or de-select events.
   (b) For selected events, increment histogram location indexed by the event pulse height.
5. Optional: Read exposure map, apply filter criteria to derive exposure time and effective area.
6. Write histogram (spectrum) file (optionally including exposure time and effective area and/or ARF file [TBD]).

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2037

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

acis_extract_image

Description: Bin (appropriately filtered) ACIS events into image for selected coordinate system. Wrapper around `acis_filter_events` and `acis_bin_events`.

Parameters:

```
X,Y coordinate system
  (CHIP, TDET, DET or sky (RA,dec))
Filter criteria for event attributes:
  TIME
  PHA
  PI
  GRADE
  STATUS
  CCDID
Bin width
X,Y ranges
```

Inputs:

```
event list
```

Outputs:

```
image
```

Processing:

1. Read parameter file and establish event attribute over which to bin image (i.e., which coord system). Evaluate filter criteria and binning parameters (spatial bin width, bin range).
2. Open and read events file.
3. Initialize image, based on coord system and bin parameters.
4. Loop over events:
   (a) Apply filter criteria to select or de-select events.
   (b) For selected events, increment image location indexed by the event coordinates.
5. Optional: Read exposure map, and bin in like manner to derive exposure time and effective area.
6. Write histogram (spectrum) file (optionally including exposure time and effective area).

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2038

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC

acis_extract_lightcurve

Description: Bin (appropriately filtered) ACIS events into light curve, and (optionally) calculate exposure time and effective area for selected parameter space. Wrapper around `acis_filter_events` and `acis_bin_events`.

Parameters:

```
Filter criteria for event attributes:
  X,Y (e.g. regions file)
    CHIP, TDET, DET or sky (RA,dec) coord systems
  GRADE
  STATUS
  CCDID
TIME bin width
TIME range
```

Inputs:

```
event list
exposure map
```

Outputs:

```
light curve (histogram of counts vs. time bin)
region exposure time
region effective area
```

Processing:

1. Read parameter file and establish event filter criteria and time binning parameters (bin width, bin range).
2. Open and read events file.
3. Initialize light curve histogram, based on event attribute and bin parameters.
4. Loop over events:
   (a) Apply filter criteria to select or de-select events.
   (b) For selected events, increment histogram location indexed by the event time.
5. Optional: Read exposure map, apply filter criteria to derive exposure time and effective area.
6. Write histogram (light curve) file (optionally including exposure time and effective area [TBD]).

Release: 4

Group: TBD

Analysis Domain: TBD

DS Tool Class: TBD

DS Tool Category: TBD

Spec Name: acis/spec/spec2039

Spec Category: TBD

Code Type: ASCDS

Code Source: ASC