ASC Data Processing Tools for ACIS
Revision 2.0

Glenn E. Allen
Chandra X-ray Center
Science Data Systems

November 1, 1999
Contents

1 Preamble
   1.1 Document and Change Control Log ........................................... iii
   1.2 Applicable Documents ......................................................... iii

2 Tool Summaries
   2.1 Level 1 Tools ................................................................. 1
   2.2 Level 2 Tools ................................................................. 1

3 Tool Descriptions
   3.1 Level 0.5 Tool ................................................................. 3
      acis_collate_events .......................................................... 3
   3.2 Level 1 Tools ................................................................. 5
      acis_format_events .......................................................... 5
      acis_correct_bias ........................................................... 11
      acis_process_events ....................................................... 13
      acis_grade_events ......................................................... 22
      acis_calc_pi ................................................................. 26
      acis_build_mask ........................................................... 28
      acis_build_badpix ......................................................... 30
   3.3 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 Preamble

1.1 Document and Change Control Log

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Tool(s)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 April 1998</td>
<td>1.0</td>
<td>...</td>
<td>First Release</td>
</tr>
<tr>
<td>29 Sept 1998</td>
<td>1.1</td>
<td>acis_format_events,</td>
<td>improved description of bias subtraction; revised CTI and event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_correct_bias,</td>
<td>event ENERGY, PI descriptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_process_events,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_grade_events, acis_calc_pi</td>
<td></td>
</tr>
<tr>
<td>07 April 1999</td>
<td>1.2</td>
<td>acis_format_events,</td>
<td>added references to CC 3x3 modes; added references to ACIS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_correct_bias,</td>
<td>flight SRR requirements for bias subtraction; clarified PHAS summation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_process_events,</td>
<td>rules; updated descriptions of tool parameters; updated status bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_grade_events</td>
<td>descriptions</td>
</tr>
<tr>
<td>01 November 1999</td>
<td>2.0</td>
<td>acis_process_events,</td>
<td>changed algorithm to compute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acis_calc_pi</td>
<td>ENERGY and PI from PHA</td>
</tr>
</tbody>
</table>

1.2 Applicable Documents

1. ACIS Data Products:
   - Level 0 to ASC Archive Interface Control Document
     link on http://space.mit.edu/ASC/docs/docs.html

2. ACIS Data Products:
   - Level 1 to ASC Archive Interface Control Document
     link on http://space.mit.edu/ASC/docs/docs.html

3. Analysis Reference Data ICD for:
   - ACIS Level 1 & 2 Pipelines and Tools
     link on http://space.mit.edu/ASC/docs/docs.html

   http://acis.mit.edu/acis/sreqj
2 Tool Summaries

2.1 Level 1 Tools

acis\_format\_events For ACIS data obtained in faint event modes (TE faint, very faint, and faint with bias; CC faint), subtract bias and overclock from pixel pulse heights (ADU). 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), subtract bias and overclock from pixel pulse heights (ADU).

acis\_process\_events Perform calculations and corrections on ACIS event attributes. Correct event PHA for charge transfer inefficiency (optional); grade ACIS events according to split geometry; sum PHA of pixels above split threshold, according to relevant grading system; convert pulse height amplitudes (PHA) in ADU to pulse-invariant (PI) values in eV; 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\_build\_evtsummary 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)

2.2 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.
3 Tool Descriptions

3.1 Level 0.5 Tool

\textbf{acis\_collate\_events}

\textbf{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.

\textbf{Parameters:}

\textbf{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

\textbf{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)

\textbf{Processing:}

1. Open parameter block file for the science run. If value of \texttt{DTYCYLE} keyword is nonzero (meaning that both primary and secondary exposure times are present in event list) then proceed with the following steps. If \texttt{DTYCYLE} = 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 \texttt{EXPTIMEA} and \texttt{EXPTIMEB}). Set \texttt{CYCLE} keyword to reflect exposure time, e.g.:

\[
\texttt{CYCLE} = 'P' \quad / \text{events are from which exps? P[rimary]/S[condary]/B[oth]}
\]

where \texttt{CYCLE='P'} corresponds to \texttt{EXPTIMEA} and \texttt{CYCLE='S'} corresponds to \texttt{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 \texttt{EXPNO} to establish expected exposure time\(^1\), based on values of parameter block keywords \texttt{DTYCYLE}, \texttt{EXPTIMEA}, and \texttt{EXPTIMEB}:
   - If \texttt{EXPNO} = (1 + \texttt{DTYCYLE}) \times N (N = 0, 1, 2,...) then exposure time is \texttt{EXPTIMEA}.
   - Otherwise, exposure time is \texttt{EXPTIMEB}.

\(^1\)Need to establish whether first exposure has \texttt{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** = \((1 + DTYCYCLE) \times N \) \( (N = 0, 1, 2, \ldots) \) 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
3.2 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), subtract bias and overclock from pixel pulse heights (ADU). 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.

**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
  - initial overclock values
  - exposure number
  - delta OC values
  - # pixels above threshold
  - # events telemetered

**Outputs:**

- bias/OC-corrected event data
- status bits
- updated bias map (TE faint w/ bias only)
- updated MTL file
- exposure statistics table

**Processing:**

1. Determine ACIS mode (one of TE faint, TE very faint, TE faint with bias, CC faint, or CC 3x3 faint)

2. Read event, bias (if nec.), bad pixel, and exposure record data.

3. For each active CCD, loop over events:
   
   (a) Subtract pixel-by-pixel bias and node- and exposure-specific overclock (tool: acis_correct_bias), according to the algorithm defined in §3.2.3.14 (TE faint, TE very faint, TE faint with
bias, CC 3x3 faint) or §3.2.3.3.12 (CC faint) of Rev I of the ACIS Flight Software Requirements Specification, http://acis.mit.edu/acis/sreqi.

(b) Update exposure statistics table using data (e.g., events telemetered and pixels above threshold) from exposure record file.

4. 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_existence (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
   ENDFWHILE
ENDIF
close input file
ENDIF
close output file
ELSE
   add err mask to status flag (diff size stacks)
ENDIF
END.

Appendix B: Description of Parameters (from help file in release as of 2/11/99)

infile - (existing fits bin table or qpoe event file)
- this is an auto parameter that is used to specify the input event file from which acis_format_events will process data.

biasfile - (a fits image file name or blank)
- This is a fits image file which is 1024x1024 pixels. It is used to bias correct data. Left blank for graded modes or faint w/ bias mode. A file name for faint te, very faint, or faint w/ bias (optional)

exrfile - (existing fits file corresponding to input event file)
- This file corresponds to the event file and contains among other data, the overclock values to use when applying overclock corrections.

outfile - (nonexistent fits bin table or qpoe event file)
- this is an auto parameter that is used to specify where the
processed events are to be written.

badpixfile - (a fits file or NONE)
- this file whose format is TBD is used to set status bits to
indicate missing or bad pixels in the event list.

outbias - (nonexistent fits image file)
- this value is used as the suffix of the name of the output
bias map created for faint w/ bias mode data.

expstatsfile - (nonexistent fits file)
- this value specifies the name of the output exposure
statistic stable created by acis_format_events. The file
also contains the dropped exposures extensions.

logfile - (nonexistent text file or 'stdout')
- This value specifies the name of the ascii text file
which acis_format_events will generate if the verbose param
(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 - (output format or redirect to pre-existing output format)
- 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.

bias_correct - (yes, no)
- This parameter serves as a flag to instruct
acis_format_events as to whether or not it should perform
bias corrections.

oc_correct - (yes, no)
- This flag allows the user to specify whether or not overclock
corrections are applied to the events.

min_init_oc - (integer 0..500)
- This value identifies the minimum acceptable value for an
initial overclock.

max_init_oc - (integer 0..500)
- This value identifies the maximum acceptable value for an
initial overclock.

min_dltat_oc - (integer -250..250)
- 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** - (integer -250..250)
- 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** - (yes, no, redirect to qpoe.par file)
- This boolean parameter instructs acis_format_events 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** - (integer or redirect to qpoe.par file)
- Allows the user to specify the qpoe page size

**qp_bucketlen** - (integer or redirect to qpoe.par file)
- Allows the user to specify the qpoe bucket length

**tempbias** - (yes, no)
- This flag is a temporary work around which causes acis_format_events to perform a simple bias correction algorithm if bias maps are unavailable.

**clobber** - (yes, no)
- This parameter instructs acis_format_events 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. (This function is currently only supported for output bias maps.)

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

**vflev1** - (output event definition string)
- 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 primarily intended for use with very faint mode data (ie. 5x5 mode data).

**cclev1** - (output event definition string)
- 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 primarily intended for continuous clocking mode data.

**verbose** - (0..5)
- Option which allows the user to request a varying level of textual output based upon the program execution. Levels range
from 0 to 5 with 0 representing no information and 5 representing as detailed a log as possible. The log is written out to the directory the function was invoked from, and is named 'runlog'.

Release: 4
Group: DA
Analysis Domain: Event
DS Tool Class: 3
DS Tool Category: Correction
Spec Name: 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**
  - initial overclock values
  - exposure number
  - delta OC values
  - # pixels above threshold
  - # events telemetered

**Outputs:**

- bias/OC-corrected event data
- updated bias map (TE faint w/ bias mode)

**Processing:** Correct for bias and overclock according to the algorithm defined in §3.2.2.3.14 (TE faint, TE very faint, TE faint with bias, CC 3x3 faint) or §3.2.3.3.12 (CC faint) of Rev I of the ACIS Flight Software Requirements Specification, http://acis.mit.edu/acis/sreqi, i.e.,

Corrected Pixel PH = raw pixel PH - pixel bias map value - (overclock - initial_overclock)

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. Subtract overclock (OC) from event pixel ADU. 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. For events that lie at node edges, special care must be taken in applying OC values for the appropriate node for each pixel.

**Release:** 4

**Group:** DA
Analysis Domain:  Event
DS Tool Class:  3
DS Tool Category:  Correction
Spec Name:  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; optionally, correct PHA for CTI (under study). See description of tool acis_grade_events.

2. Calculate ENERGY and PI from event PHA based on CCD gain. See description of tool acis_calc_pi.

3. Calculate tiled detector and 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. Randomize coordinates by adding a uniform, random offset of between −0.5 and +0.5 pixels to CHIPX and CHIPY, and apply the chip to detector coordinate transformation for each event. (For CC mode events, CHIPY is set to 512 prior to determining detector coordinates.) Write the resulting tiled detector and detector coordinates to the event list.

4. Calculate world (celestial) coordinates: apply aspect solution to 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_instrument_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_instrument_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)
```
Appendix B: Description of Parameters (from help file in release as of 2/11/99)

infile - (existing event file/stack - either qpoe or fits)
- this is an auto parameter that is used to specify the input
  event file from which acis_process_events will process data.
  If the file is in fits format, the actual event information
  must be in an extension named 'EVENTS'. Additional items
  are extracted from keywords in the primary and EVENTS
  extensions.

outfile - (nonexistent event file- either qpoe or fits)
- this is an auto parameter that is used to specify where the
  processed events are to be written.

acofffile - (existing .FITS file or NONE)
- offset file used to compensate for spacecraft movements
during an observation

alignmentfile - (existing alignment .FITS file or NONE)
- this is an auto parameter which is used to provide the tool
  with values used to adjust the mirror position via sim
  alignment (flight) or dither (xrcf) values.

logfile - (nonexistent text file or 'stdout')
- This value specifies the name of the ascii text file
  which acis_process_events will generate if the verbose param
  (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)

obsfile - (existing observation parameter .PAR file or NONE)
- This value specifies the name of the observation parameter
  file to seed the output event file header with. If the value
  is not "NONE", the keywords from the specified file are
  copied to the output file’s header.

gainfile - (existing gain correction .FITS file or NONE)
- This value specifies the name of the gain correction file
  to use when calculating pi/performing gain correction. The
  calculate_pi parameter (see below) must be set to perform
  gain correction.
gradefile - (existing grade .FITS file or NONE)
- This value specifies the name of the grading file to use to perform grading. The file is also used to determine how to sum corner pixels for event pulse height summations.

threshfile - (.FITS file or NONE)
- This optional parameter specifies the name of the file to use to specify ccdid and ccd node specific split threshold values. If set to 'NONE' the value specified in the spthresh parameter (see below) is used as the default value for all ccds.

eventdef - (output format or redirect to pre-existing output format)
- 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.

doevtgrade - (yes, no)
- option to calculate and output the flight grade and pulse height sum of events in the input qpoe file.

spthresh - (0..32767)
- the user defined split threshold value used in determining centroid calculations as well as flight grade and pulse height sum values. (This is the default threshold value which is used if the threshfile parameter (see above) is set to 'NONE').

time_offset - (real number)
- offset which needs to be added to the event file to synch it up with the alignment data.

docentroid - (yes, no)
- option to adjust the location of the local maxima based upon weighted values of surrounding pixel pulse heights above a threshold value.

calculate_pi - (yes, no)
- option to calculate pi/perform gain correction based on the values in the gain file (see above) which must also be provided. The computed pulse invarience is only written to the output event file if it is specified in the output file's eventdef ('f:energy,s:pi').

pi_bin_width - (1..1000)
- This value specifies the scale in eV for the pi column units.

pi_num_bins - (256..32767)
- This value specifies the range of the pi column. Each unit in the range represents a range of pi_bin_width (see above).
randompha - (yes, no)
- This parameter determines whether a random offset is added to the integer PHA values of the events.

tstart - (TSTART or some other keyword name)
- The name of the header keyword containing the time of the first event in this file. It is used as the lowerbound of a temporal filter on the event file.

tstop - (TSTOP or some other keyword name)
- The name of the header keyword containing the time of the last event in this file. It is used as the upperbound of a temporal filter on the event file.

qp_internals - (yes, no, redirect to qpoe.par file)
- This boolean parameter instructs acis_process_events 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 - (integer or redirect to qpoe.par file)
- allows the user to specify the qpoe page size

qp_bucketlen - (integer or redirect to qpoe.par file)
- allows the user to specify the qpoe bucket length

verbose - (0..5)
- Option which allows the user to request a varying level of textual output based upon the program execution. Levels range from 0 to 5 with 0 representing no information and 5 representing as detailed a log as possible. The log is written out to the directory the function was invoked from, and is named 'runlog'.

stop - (chip, tdet, det, sky)
- The end of the coordinate transformations. This determines the extent of the coordinate transformations that are executed by acis_process_events. It should generally be set to sky.

instrume - (acis)
- This specifies the instrument that the data was collected with. It should be set to acis.

telecop - (name of telescope parameter file or blank)
- This tells what telescope parameter file to utilize

random - (yes, no)
- This parameter controls whether a randomization factor is added into coordinates when they are written out to
compensate for any aliasing which may occur due to float to integer truncation.

rand_seed  - (0 or positive value)
- This value specifies the random seed which is to be applied to randomizations. A value of 0 causes the code to utilize a time based pseudo-random seed. A non-zero value should produce identical results on multiple runs of the same data.

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

cclev1     - (output event definition string)
- 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 primarily intended for continuous clocking mode data.

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

NOTE: Bits 01,02,03 are set in acis_process_events; bits 04,06,07,08,09,10 and 12,13,14 are set in acis_format_events. Bits 00,05,11 not implemented as of 8/97.

<table>
<thead>
<tr>
<th>Status Mask</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1_STS_GOOD_EVENT</td>
<td>0x0000</td>
</tr>
<tr>
<td>L1_STS_BAD_POS</td>
<td>0x0001</td>
</tr>
<tr>
<td>L1_STS_WM_BAD.VAL</td>
<td>0x0002</td>
</tr>
<tr>
<td>L1_STS_PIX_RANGE</td>
<td>0x0004</td>
</tr>
<tr>
<td>L1_STS_WM_BAD_SUM</td>
<td>0x0008</td>
</tr>
<tr>
<td>L1_STS_WM_BAD_PIX</td>
<td>0x0010</td>
</tr>
<tr>
<td>L1_STS_ADJ_BAD_PIX</td>
<td>0x0020</td>
</tr>
<tr>
<td>L1_STS_BIAS_BAD_PIX</td>
<td>0x0040</td>
</tr>
<tr>
<td>L1_STS_BIAS_MISSING</td>
<td>0x0080</td>
</tr>
<tr>
<td>L1_STS_BIAS_PARITY</td>
<td>0x0100</td>
</tr>
<tr>
<td>L1_STS_OC_MISSING</td>
<td>0x0200</td>
</tr>
<tr>
<td>L1_STS_OC_RANGE</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
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 maximum (highest PHA value in the island) or which have center pixel PHA values that fall below split threshold. These pixels, and/or side and corner pixels that equal or exceed both the center pixel and split threshold, are included in total event PHA.
- 02 - This bit is set to 1 for any event which has one or more of its individual island pixels > 4095 after correction for overclock and bias.
- 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. These pixels are not included in total event PHA.

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. These pixels are not included in total event PHA.
- 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. These pixels are not included in total event PHA.
- 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. These pixels are not included in total event PHA.
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\geq oc\_val \geq 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 corner 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 do 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: 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.

Parameters:

split thresholds (CCD/node specific) (ARD)
number of overclock pairs

Inputs:

event pixels
grades table (flight bitmap to ASCA/ACIS/USER) (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, graded, 3x3 faint, or 3x3 graded).

2. OPTIONAL: If events obtained in any of the timed exposure\(^2\) faint modes, correct all 9 (25, for very faint mode) 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/ASCA/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:

<table>
<thead>
<tr>
<th>Node</th>
<th>min(CHIPX)</th>
<th>max(CHIPX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>1</td>
<td>257</td>
<td>512</td>
</tr>
<tr>
<td>2</td>
<td>513</td>
<td>768</td>
</tr>
<tr>
<td>3</td>
<td>769</td>
<td>1024</td>
</tr>
</tbody>
</table>

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

\(\text{\footnotesize \text{\color{red}{\text{\textsuperscript{2}No CTI correction algorithm is TBD for continuous clocking readout mode.}}}}\)
(c) Determine parallel and serial (alternatively, X and Y, or CCD column and row) charge loss from CTI coefficients as follows:

\[ C_X = P_a(n) + P_c(n) \times (\text{PHA}_a(n)) \]

\[ C_Y = S_a(n) + S_c(n) \times (\text{PHA}_s(n)) \]

where \( C_X \) and \( C_Y \) are the parallel and serial charge loss; \( P_a, P_b, P_c \) and \( S_a, S_b, S_c \) are the parallel and serial CTI coefficients, respectively; \( n \) is the node; and \( \text{PHA} \) is the latest (corrected) event PHA (equal to PHA', below). Initially, set \( \text{PHA} = \text{PHA}_0 \), where \( \text{PHA}_0 \) is the raw (original, uncorrected) event PHA.

(d) Determine column (X) offset from readout node (= number of serial transfers):

\[
\begin{align*}
\text{if (node eq 0 or node eq 2) then } nx &= \text{CHIPX} - (\text{node} \times 256) + n_{oc} \\
\text{if (node eq 1 or node eq 3) then } nx &= ((\text{node} + 1) \times 256) - \text{CHIPX} + n_{oc}
\end{align*}
\]

where \( n_{oc} \) is \( 2 \times \text{number of overclock pairs} \). The row (Y) offset from readout node (number of parallel transfers) is just given by \( n_Y = \text{CHIPY} \).

(e) Calculate CTI-corrected PHA (= PHA') for each of the 9 elements in the 3x3 pixel event island as follows:

\[
\begin{align*}
\text{PHAS'}[0] &= \text{round}(\text{PHAS}[0] / (1.0 - (nx-1)*C_X - (ny-1)*C_Y) + 0.5) \\
\text{PHAS'}[1] &= \text{round}(\text{PHAS}[1] / (1.0 - nx*C_X - (ny-1)*C_Y) + 0.5) \\
\text{PHAS'}[2] &= \text{round}(\text{PHAS}[2] / (1.0 - (nx+1)*C_X - (ny-1)*C_Y) + 0.5) \\
\text{PHAS'}[3] &= \text{round}(\text{PHAS}[3] / (1.0 - (nx-1)*C_X - ny*C_Y) + 0.5) \\
\text{PHAS'}[4] &= \text{round}(\text{PHAS}[4] / (1.0 - nx*C_X - ny*C_Y) + 0.5) \\
\text{PHAS'}[5] &= \text{round}(\text{PHAS}[5] / (1.0 - (nx+1)*C_X - ny*C_Y) + 0.5) \\
\text{PHAS'}[6] &= \text{round}(\text{PHAS}[6] / (1.0 - (nx-1)*C_X - (ny+1)*C_Y) + 0.5) \\
\text{PHAS'}[7] &= \text{round}(\text{PHAS}[7] / (1.0 - nx*C_X - (ny+1)*C_Y) + 0.5) \\
\text{PHAS'}[8] &= \text{round}(\text{PHAS}[8] / (1.0 - (nx+1)*C_X - (ny+1)*C_Y) + 0.5)
\end{align*}
\]

An analogous set of 25 expressions holds for very faint mode.

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) and the ACIS Flight Software Requirements Specification (§3.2.2.3.16, 3.2.3.3.14). 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.

```
| +---------+---------+---------+---------+---------+---------+---------+---------+---------+
| | x | - | - | - | row N+1          |
| +---------+---------+---------+---------+---------+---------+---------+---------+---------+
| | - | X | - | - | row N               |
| +---------+---------+---------+---------+---------+---------+---------+---------+---------+
| | x | - | - | - | row N-1             |
| +---------+---------+---------+---------+---------+---------+---------+---------+---------+
| ^ ^ ^     | direction     |
| |         | v            |
| |         | +-------- col M+1 |
| |         | +-------- col M    |
| |         | +-------- col M-1  |
| 0,0       |               |
```

A B C D <--- output nodes

Arrows show directions of serial clocking

For continuous clocking (CC) 1x3 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). The bitmap assignments for CC 3x3 faint mode data are identical to those in TE faint modes.

(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 value of the keyword CORNERS in the grade system analysis reference data file:

-1 => never include corners in total PHA
0 => always include corners
1 => flight software (ACIS) convention (see ACIS flight s/w requirements spec, Sec 3.2.2.3.16)
2 => ASCA convention (see http://heasarc.gsfc.nasa.gov/docs/asca/sis_grade.gif)

In the ASCA system corner pixel PHAs are generally excluded, whereas in the ACIS system, corner pixel PHAs are included if adjacent to at least one “side” pixel at or above split threshold.

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; see http://space.mit.edu/ASC/docs/grades.ps.gz. See Analysis Reference Data ICD for specification 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: spec23

Spec Category: Correction

Code Type: ASCDS

Code Source: ASC
**acis_calc_pi**

**Description:** Convert the pulse height amplitudes (PHA) of events in ADU to ENERGY in eV and pulse-invariant (PI) values in channels.

**Parameters:**

- PI bin width (default: 14.6 eV)
- PI array size (default: 1024)
- Randomize PHAs (default: yes)

**Inputs:**

- event list
- Analysis Reference Data (ARD) gain_map table (i.e. acisDyyyy-mm-ddgainNmmn_mnn.fits)

**Outputs:**

- Updated event list (with ENERGY and PI columns)

**Processing:**

For each event in an event list:

1. Verify that the PHA column is present in the event list.

2. Convert the value of PHA for an event from an integer number to a real number by adding an offset that is randomly chosen from a uniform distribution between $-0.5$ and $+0.5$. The use of the random offset is optional and controlled by a “yes/no” parameter. The default condition for this parameter is “yes” (i.e. use the offset).

3. Calculate the value of the ENERGY of an event that corresponds to the value of the PHA of the event using the vectors named ENERGY and PHA in the gain ARD file. Select the gain ARD file that is appropriate for the nominal temperature of the focal plane during the observation (in a manner that is TBD). (See [http://space.mit.edu/ASC/docs/docs.html](http://space.mit.edu/ASC/docs/docs.html) for a description of the gain ARD files.) Since the gain depends on the position of an event on a chip, the vectors ENERGY and PHA are tabulated in the gain ARD region-by-region for each chip. To compute the value of the ENERGY of an ACIS event:

   i. Find the vectors ENERGY and PHA of the row in the gain ARD file that satisfies the conditions that

   \[
   \text{chip id} = \text{CCD\_ID},
   \text{CHIPX\_MIN} \leq \text{CHIPX} \leq \text{CHIPX\_MAX}, \text{and}
   \text{CHIPY\_MIN} \leq \text{CHIPY} \leq \text{CHIPY\_MAX},
   \]

   where “chip id” is the chip on which the event was detected, CHIPX and CHIPY specify the location on the chip at which the event was detected, and CCD\_ID, CHIPX\_MIN, CHIPX\_MAX, CHIPY\_MIN, and CHIPY\_MAX are the names of columns in the gain ARD file.
ii. Find the two non-zero (real!) values of $\text{PHA}_i$ and $\text{PHA}_{i+1}$ in the vector $\text{PHA}$ such that

$$0 < \text{PHA}_i \leq \text{PHA} < \text{PHA}_{i+1},$$

where $\text{PHA}$ is the pulse height amplitude of the event.

iii. Set the $\text{ENERGY}$ of the event to be

$$\text{ENERGY} = \frac{\text{PHA} - \text{PHA}_i}{\text{PHA}_{i+1} - \text{PHA}_i} \left( \text{ENERGY}_{i+1} - \text{ENERGY}_i \right) + \text{ENERGY}_i,$$

where $\text{ENERGY}_i$ and $\text{ENERGY}_{i+1}$ are the non-zero values in the vector $\text{ENERGY}$ that correspond to the values of $\text{PHA}_i$ and $\text{PHA}_{i+1}$, respectively.

This formula is valid if the value of the PHA of the event satisfies

$$0 < \text{PHA}_1 \leq \text{PHA} < \text{PHA}_n,$$

where $\text{PHA}_1$ and $\text{PHA}_n$ are the smallest and largest, respectively, non-zero values of PHA in the vector $\text{PHA}$. If $0 < \text{PHA} < \text{PHA}_1$,

$$\text{ENERGY} = \max \left( \frac{\text{PHA} - \text{PHA}_1}{\text{PHA}_2 - \text{PHA}_1} \left( \text{ENERGY}_2 - \text{ENERGY}_1 \right) + \text{ENERGY}_1, 0 \right).$$

If $\text{PHA} \geq \text{PHA}_n$,

$$\text{ENERGY} = \frac{\text{PHA} - \text{PHA}_n}{\text{PHA}_{n-1} - \text{PHA}_n} \left( \text{ENERGY}_n - \text{ENERGY}_{n-1} \right) + \text{ENERGY}_{n-1}.$$

4. Compute the value of the PI of an event by dividing the value of the $\text{ENERGY}$ of the event by the PI bin width parameter (e.g. 14.6 eV by default) and by rounding the result to the nearest integer in the range from 1 to $n$, where $n$ is the number of PI bins (e.g. 1024 by default). The value of PI may have a value of zero in the special case that PHA has a value of zero. In this case, also set the value of $\text{ENERGY}$ to zero.

5. Write the values of the $\text{ENERGY}$ and PI to the output event file.

**Release:** 4  
**Group:** DA  
**Analysis Domain:** Event  
**DS Tool Class:** 3  
**DS Tool Category:** Correction  
**Spec Name:** 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:
   
   \[
   \text{FIRSTROW} = \text{STARTROW} + 1 \\
   \text{NROWS} = \text{STARTROW} + \text{ROWCNT} + 1
   \]

   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:
   
   \[
   \begin{align*}
   \text{WINDOW} & \text{ (running value from 1 up to, but not exceeding, 6 for each CCD)} \\
   \text{LL_CHIPX} &= \text{LL_CCDX} + 1 \\
   \text{LL_CHIPY} &= \text{LL_CCDY} + 1 \\
   \text{UR_CHIPX} &= \text{LL_CCDX} + \text{CCDCOL} + 1 \\
   \text{UR_CHIPY} &= \text{LL_CCDY} + \text{CCDROW} + 1
   \end{align*}
   \]
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: 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 Bad 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 ingest 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 ingest 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:

<table>
<thead>
<tr>
<th>TIME</th>
<th>CHIPX</th>
<th>CHIPIY</th>
<th>ORIGIN</th>
</tr>
</thead>
</table>

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 extension of output Bad Pixel table file.

 Release: 4
 Group: TBD
 Analysis Domain: TBD
 DS Tool Class: TBD
 DS Tool Category: TBD
 Spec Name: spec2032
 Spec Category: TBD
 Code Type: ASCDS
 Code Source: ASC
3.3 Level 2 Tools

\texttt{acis\_filter\_events}

\textbf{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 \texttt{data\_copy}.

\textbf{Parameters}:

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

\textbf{Inputs}:

`Raw’’ event list

\textbf{Outputs}:

Filtered event list
Filter criteria (as header keywords)

\textbf{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.

\textbf{Release}: 4

\textbf{Group}: TBD

\textbf{Analysis Domain}: TBD

\textbf{DS Tool Class}: TBD

\textbf{DS Tool Category}: TBD

\textbf{Spec Name}: spec2035

\textbf{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 attribute 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.

**Release:** 4

**Group:** TBD

**Analysis Domain:** TBD

**DS Tool Class:** TBD

**DS Tool Category:** TBD

**Spec Name:** 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: spec037
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: 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: spec2039
Spec Category: TBD
Code Type: ASCDS
Code Source: ASC