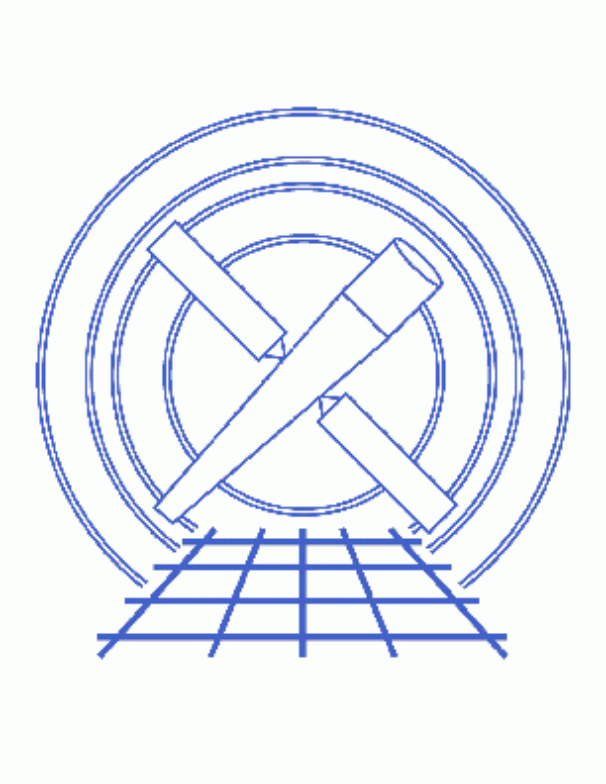
Apply an ACIS Gain Map



CIAO 3.4 Science Threads

Table of Contents

Apply an ACIS Gain Map

CIAO 3.4 Science Threads

Overview

Last Update: 21 May 2007 − need to set `stop=none` if aspect solution is not provided

Synopsis:

The gainfile is used to compute the ENERGY and PI of an event from the PHA value; see the PI dictionary entry for more information. Each of the values is stored in a column of the same name in the event file. Using the correct gainmap is especially important if one is interested in ENERGY, PI, or the data is for a grating observation. Later in the analysis, the FEF and OSIP files that correspond to the gain map should be used to create an RMF for the data.

Purpose:

To generate a new level=2 event file with the proper gain map applied. If you have already run the Apply the ACIS CTI Correction thread, the newest available gain map has been to the data; it ***is not necessary*** to run this thread as well.

Read this thread if:

you are working with any ACIS observation (imaging or grating) that is affected by the Calibration Updates.

Note: if you are working with imaging data, it is important that the calibration applied to the event file is consistent with the RMF tool chosen. Specifically, a different gain file must be used when running `acis_process_events`; more information is available in the Creating ACIS RMFs why topic. ***It is recommended that users apply the default gain file to the data, and use `mkacisrmf` to create the imaging RMF file.***

Calibration Updates:

Get Started shows how to check the CALDBVER and temperature of your observation.

- ***CALDB v3.3.0*** ***(18 Dec 2006):*** A new set of gain files for −120 data have been added to the CALDB. Only calibration for the BI chips (S1, S3) has changed in this file; calibration for the FI chips is identical to the v5 file. The How CIAO 3.4 and CALDB 3.3.0 Affect Your Analysis section of the CIAO release notes explains how the files will affect your analysis.
- ***CALDB v3.2.1*** ***(15 Dec 2005):*** The gain for the back−illuminated (BI) chips − ACIS−S1 and S3 − has been upgraded to match the BI gains from the acisD2000−01−29gain_ctiN0005.fits file. See How CALDB 3.2.1 Affects Your Analysis for details.
- ***CALDB v3.2.0*** ***(21 Nov 2005):*** New gain files for −120 data have been added to the CALDB; see the CIAO 3.3 release notes for information on how they will affect your analysis.

- ***CALDB v3.0.0*** *(15 Dec 2004):* New default gain file (`acisD2000-01-29gain_ctiN0003.fits`).
- ***CALDB v2.7*** *(7 Aug 2001):* New gainfiles for analyzing −120 degree observations taken on the S3 chip.
- ***CALDB v2.2*** *(15 Feb 2001):* All the gainfiles were updated in this release.

Related Links:

- Analysis Guide: ACIS Data Preparation
- Apply the ACIS CTI Correction thread: it is strongly recommended that you reprocess imaging data with this new calibration which became part of standard data processing in DS 6.11. The CTI correction is on by default (`apply_cti=yes`) in `acis_process_events`.
- Corrections for time−dependence of ACIS gain: describes the gain change correction algorithm which was derived for the CTI−corrected data in the FI chips and the uncorrected data in S3. This correction will be implemented in `acis_process_events` in a future release of CIAO; it can be applied now via a C program available from the software exchange page.

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

Get Started

Sample ObsID used: 1838 (ACIS−S, G21.5−09)

File types needed: evt1; flt1; bpix1

If you created a new bad pixel file by running the Create a New ACIS Bad Pixel File: Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread, use that file in this analysis. Otherwise, use the `bpix1.fits` file from the Archive.

Check the CALDBVER keyword in the header

```
unix% dmkeypar acisf01838_000N001_evt1.fits CALDBVER echo+
 1.4
```

Also, determine if this is −120 degree S3 data (as explained in the Overview):

```
unix% dmkeypar acisf01838_000N001_evt1.fits FP_TEMP echo+
153.446014

unix% dmkeypar acisf01838_000N001_evt1.fits DETNAM echo+
ACIS-012367
```

See this FAQ for more information on checking the temperature of your observation. The S3 chip is ccd_id=7, as shown in Figure 6.1 of the POG.

Since this is −120 degree S3 data which was processed with a CALDBVER lower than 3.2.0, complete this thread in its entirety.

Related acis_process_events threads

There are other threads that should be considered, since they may affect how `acis_process_events` is run. ***The Create a New Level=2 Event File thread shows how to combine all of these options into a single run of `acis_process_events`.***

- Apply the Time−Dependent ACIS Gain Correction
- Remove Pixel Randomization
- Apply/Remove PHA Randomization
- Apply the ACIS CTI Correction

Generate A New Level=1 Event File

Determine the eventdef parameter

The `eventdef` parameter specifies the names and data types of the columns in the output event data file. Four predefined strings are included in the parameter file for acis_process_events:

READMODE	DATAMODE	event mode	eventdef string
TIMED	(V)FAINT	timed exposure (very) faint	stdlev1
TIMED	GRADED	timed exposure graded	grdlev1
CONTINUOUS	CC(33)_FAINT	continuous clocking (3x3) faint	cclev1
CONTINUOUS	CC(33)_GRADED	continuous clocking (3x3) graded	ccgrdlev1

If you are unsure of the event mode of your observation, the information can be found in the `READMODE` and `DATAMODE` values stored in the file header:

```
unix% dmkeypar acisf01838_000N001_evt1.fits READMODE echo+
TIMED

unix% dmkeypar acisf01838_000N001_evt1.fits DATAMODE echo+
FAINT
```

This is a timed exposure faint observation, so the proper `eventdef` parameter is "stdlev1." The full parameter syntax of each eventdef string may be found in plist acis_process_events.

Run acis_process_events

Running this tool with the SDP level=1 event file as the input will produce a *new* level=1 event file that has the latest CALDB applied, meaning that the newest gain map will be picked up. Since the CTI and Time−Dependent Gain corrections are on by default, they will both be applied (when possible).

```
unix% punlearn acis_process_events
unix% pset acis_process_events infile=acisf01838_000N001_evt1.fits
```

```
unix% pset acis_process_events outfile=acis_1838_new_evt1.fits
unix% pset acis_process_events badpixfile=acis_1838_new_bpix1.fits
unix% pset acis_process_events eventdef=")stdlev1"
unix% pset acis_process_events stop=none
unix% acis_process_events
Input event file or stack (acisf01838_000N001_evt1.fits):
Output event file name (acis_1838_new_evt1.fits):
aspect offset file ( NONE | none | <filename>) (NONE):
```

It is important to note the unusual syntax of the `eventdef` parameter; the tool will not access the predefined string if the leading ")" is missing (see example 6 of `ahelp parameter`).

The content of the parameter file may be checked using plist acis_process_events.

You may see a warning about the number of event islands that contain one or more bad pixels:

```
# acis_process_events (CIAO 3.4): The following error occurred 26941
  times: dsAFEBADPCNTERR -- WARNING: Event island contains 1 or more bad pixels.
```

It is explained in this FAQ and may be ignored.

Generate A New Level=2 Event File

If you are working with grating data, you should proceed to the Obtain Grating Spectra from HETG/ACIS−S Data thread or the Obtain Grating Spectra from LETG/ACIS−S Data thread at this point to generate the correct level=1.5 and level=2 files. For non−grating data, continue with the following steps.

Apply grade/status filters

Filter for bad grades (using ASCA grades) and for a "clean" status column (ie all bits set to 0):

```
unix% punlearn dmcopy
unix% dmcopy "acis_1838_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
      acis_1838_flt_evt1.fits
```

Apply GTI filters

The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied. Simultaneously, an unnecessary column is eliminated from the output:

```
unix% punlearn dmcopy
unix% dmcopy "acis_1838_flt_evt1.fits[EVENTS][@acisf01838_000N001_flt1.fits][cols -phas]" \
      acis_1838_evt2.fits
```

Be sure to include the ***@ symbol*** in the filter expression; the command will not be executed properly if it is omitted.

Summary

This thread is complete; the new level=2 event file is named acis_1838_evt2.fits.

```
Parameters for /home/username/cxcds_param/acis_process_events.par


#-------------------------------------------------------------------------
#
#  acis_process_events.par- Parameter file for acis_process_events program
#
#-------------------------------------------------------------------------
        infile = acisf01838_000N001_evt1.fits  Input event file or stack
       outfile = acis_1838_new_evt1.fits Output event file name
    acaofffile = NONE             aspect offset file ( NONE | none | <filename>)
    (apply_cti = yes)             Apply CTI adjustment?
  (apply_tgain = yes)             Apply time-dependent gain adjustment?
(alignmentfile = )acaofffile -> NONE) sim/fam alignment file ( NONE | none | <filename>)
      (obsfile = NONE)            obs.par file for output file keywords ( NONE | none | <filename>)
      (geompar = geom)            Parameter file for Pixlib Geometry files
      (logfile = stdout)          debug log file ( STDOUT | stdout | <filename>)
    (gradefile = CALDB)           grade mapping file ( NONE | none | CALDB | <filename>)
     (gainfile = CALDB)           acis gain file ( NONE | none | CALDB | <filename>)
   (badpixfile = acis_1838_new_bpix1.fits) acis bad pixel file ( NONE | none | <filename>)
   (threshfile = CALDB)           split threshold file ( NONE | none | CALDB | <filename>)
      (ctifile = CALDB)           acis CTI file ( NONE | none | CALDB | <filename>)
    (tgainfile = CALDB)           gain adjustment file ( NONE | none | CALDB | <filename>)
     (eventdef = )stdlev1 -> {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,s:phas,l:p
f:energy,l:pi,s:fltgrade,s:grade,x:status}) output format definition
   (doevtgrade = yes)             Determine event flight grade?
 (check_vf_pha = no)              Check very faint pixels?
(calc_cc_times = no)              Estimate the times of arrival for CC-mode observation?
        (trail = 0.027)           Trail fraction
     (spthresh = 13)              Default split threshold level (overridden by values in threshfile)
  (time_offset = 0)               Offset to add to event time field to synch w/ fam data
   (docentroid = no)              Determine pixel centroid for coord. conversion?
 (calculate_pi = yes)             perform pha->pi conversion? (requires gain file)
 (pi_bin_width = 14.6)            Width of Pi bin in eV
  (pi_num_bins = 1024)            Number of values to bin energy into
 (max_cti_iter = 15)              Maximum iterations for the CTI adjustment of each event
 (cti_converge = 0.1)             The convergence criterion for each CTI-adjusted pixel in adu
       (tstart = TSTART)          header key containing start/default time value
        (tstop = TSTOP)           header key containing time of last event
      (clobber = no)              Overwrite output event file if it already exists?
      (verbose = 0)               level of debug detail (0=none, 5=most)
         (stop = none)            end transformations at [chip,tdet,det,tan,sky,none]
     (instrume = acis)            axaf instrument- used for instrument parameter file
    (rand_seed = 1)               random seed (for pixlib), 0 = use time dependent seed
     (rand_pha = yes)             Randomize the pha value used in gain calculations
(rand_pix_size = 0.5)             pixel randomization width (-size..+size) 0=no randomization
      (stdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,s:phas,l:pha,l:pha_ro,
l:pi,s:fltgrade,s:grade,x:status}) TE faint modes event definition string
      (grdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,l:pha,l:pha_ro,s:corn_
l:pi,s:fltgrade,s:grade,x:status}) TE graded event format definition string
       (cclev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,f:sky_1d,s:phas,l:pha,
f:energy,l:pi,s:fltgrade,s:grade,x:status}) CC faint event format definition string
    (ccgrdlev1 = {d:time,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,f:sky_1d,l:pha,l:pha_r
```

```
f:energy,l:pi,s:fltgrade,s:grade,x:status}) cc graded event format definition string
         (mode = ql)
```

History

03 Jan 2005	updated for CIAO 3.2: minor changes to parameter files; new default gain file (CALDB 3.0.0); use ACIS bad pixel file (`badpixfile` parameter); removed reference to running on a level=2 event file
01 Feb 2005	added note about "Event island contains 1 or more bad pixels" warning
20 Jun 2005	CIAO 3.2.2 patch: minor `acis_process_events` parameter change (default value of `threshfile` is `CALDB` instead of `NONE`)
12 Dec 2005	updated for CIAO 3.3: new gain files in CALDB 3.2.0; output filenames include ObsID
14 Jun 2006	late update: new gain files in CALDB 3.2.1
18 Dec 2006	updated for CIAO 3.4: new calibration files in CALDB 3.3.0; removed use of "rand_pha=no" in `acis_process_events`, as most users should keep the PHA randomization (see the Apply/Remove PHA Randomization thread); CIAO version in warnings
21 May 2007	need to set `stop=none` if aspect solution is not provided

URL: http://cxc.harvard.edu/ciao/threads/acisgainmap/ Last modified: 21 May 2007