Create a New Level=2 Event File

CIAO 3.4 Science Threads

Table of Contents

Create a New Level=2 Event File

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 − updated for CIAO 3.4: CIAO version in errors and warnings

Synopsis:

The structure of this thread is presented as a series of questions and answers, the latter of which are links that take you further down the analysis tree. ***Following this thread from the top down will not produce correct results.*** In order to avoid compiling too much information into one place, only basic explanations are included with the steps given here. Whenever possible, links are provided to other CIAO threads which contain more detailed information.

Purpose:

To generate a new level=2 event file for all possible grating/detector combinations.

Read this thread if:

you would like to create a new level=2 event file for ***ANY*** Chandra observation.

Related Links:

- Analysis Guide: ACIS Data Preparation
- Analysis Guide: HRC Data Preparation

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

Get Started

Since this thread is meant to represent the most generic case, it is not written with a specific ObsID. Files are referred to by the detector and type −− `acisf01843_000N001_evt1.fits` becomes ***acis_evt1.fits***, `hrcf01557_000N001_std_flt1.fits` becomes ***hrc_std_flt1.fits***, and so on.

Furthermore, it is assumed that all relevant files are in the same working directory.

Proceed to the next section to begin.

ACIS or HRC?

Question 1: which detector was used for your observation? (It does not matter yet whether or not a transmission grating was also used.)

- ACIS
- HRC

ACIS Observations

Data Preparation

If you have not already done so, it is recommended that you remove the acis_detect_afterglow correction and create a new bad pixel file, if necessary. For more information, see the ACIS Data Preparation Analysis Guide.

Creating a New Level=1 File

Running `acis_process_events` with the level=1 event file as the input will produce a *new* level=1 event file that has the latest CALDB applied. This means that the newest gain map will be picked up. In addition, you are applying the pixel and PHA randomization and the ACIS CTI correction; all of these are part of standard data processing (SDP) and are "on" by default in `acis_process_events`.

The time−dependent gain correction is also turned on by default, so it will be included in the processing. The time−dependent gain and CTI corrections are only applicable to −120 C data, but it does not hurt to leave them on for other temperatures; `acis_process_events` will simply issue a warning in that case.

See the following threads for more information:

- Apply an ACIS Gain Map
- Apply the Time−Dependent ACIS Gain Correction
- Apply the ACIS CTI Correction
- Remove Pixel Randomization
- Apply/Remove PHA Randomization
- Clean ACIS Background in VFAINT Mode (not applied in this example)

Create a new bad pixel file

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.

Determine the eventdef

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 acis_evt1.fits READMODE echo+
TIMED

unix% dmkeypar acis_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.

Set the parameters

```
unix% punlearn acis_process_events
unix% pset acis_process_events infile=acis_evt1.fits
unix% pset acis_process_events outfile=acis_new_evt1.fits
unix% pset acis_process_events badpixfile=acis_new_bpix1.fits
unix% pset acis_process_events acaofffile=pcad_asol1.fits
unix% pset acis_process_events eventdef=")stdlev1"
```

It is important to note several things here:

- In some cases there will be more than one pcad_asol1.fits file for an observation. ***All*** the files must be input to the `acaofffile` parameter ***in chronological order*** (the time is in the filename, so "`ls`" lists them in order), either as a comma−separated list or as a stack (see `ahelp stack` for more information).
- If you are working with a VFAINT observation and wish to include the background cleaning, add:

```
unix% pset acis_process_events check_vf_pha=yes
```

Execute the tool

```
unix% acis_process_events
Input event file or stack (acis_evt1.fits):
Output event file name (acis_new_evt1.fits):
aspect offset file ( NONE | none | <filename>) (pcad_asol1.fits):
```

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.

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

Question 2: is your observation imaging or grating data?

- Imaging
- Grating

Imaging Observations

There are two filtering steps that need to be completed for all ACIS imaging observations:

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

   ```
   unix% punlearn dmcopy
   unix% dmcopy "acis_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
         acis_flt_evt1.fits
   ```
2. The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied. The GTIs are contained in the `flt1.fits` file. Simultaneously, unnecessary columns are eliminated from the output:

   ```
   unix% punlearn dmcopy
   unix% dmcopy "acis_flt_evt1.fits[EVENTS][@acis_flt1.fits][cols −phas]" \
         acis_evt2.fits
   ```
   Be sure to include the ***@ symbol*** in the filter expression; the command will not be executed properly if it is omitted.

Finally, there is the option to run `destreak`, which can improve the quality of your data. This tool became part of SDP at software version DS 7.3.0. See the Destreak the ACIS−S4 Chip thread for more information.

The `ccd_id` parameter should always be set to 8 (the default) to ensure that the destreak algorithm will be applied to the ACIS−S4 chip only:

```
unix% punlearn destreak
unix% pset destreak infile=acis_evt2.fits
unix% pset destreak outfile=acis_dstrk_evt2.fits
unix% destreak
Input dataset/block specification (acis_evt2.fits):
Output dataset/block specification (acis_dstrk_evt2.fits):
```

The contents of the parameter file may be checked using plist destreak.

The thread is now complete; acis_dstrk_evt2.fits is a level=2 event file with the latest calibration products applied.

Grating Observations

There are three CIAO threads that deal specifically with generating a level=2 event file for an ACIS grating observation. Please choose the appropriate one, depending on which transmission grating was used; note there is a separate thread for analysing multiple sources:

Single source:

- Obtain Grating Spectra from HETG/ACIS−S Data
- Obtain Grating Spectra from LETG/ACIS−S Data

Multiple sources:

- Obtain Grating Spectra for Multiple Sources − ACIS (applies to both HETG and LETG)

Once you leave this thread, you ***do not*** need to return to complete any further steps.

HRC Observations

Running `hrc_process_events` with the level=1 event file as the input will produce a *new* level=1 event file with the degap, `AMP_SF`, and tapringing corrections applied.

See the following threads for more information:

- HRC AMP_SF Correction and Reducing Tap−Ringing Distortions
- HRC−S Degap Correction
- HRC−I Degap Correction

The `instrume` parameter, which specifies the instrument with which the data was collected, first needs to be determined:

```
unix% dmkeypar hrc_A_evt1.fits detnam echo+
HRC−I

unix% dmkeypar hrc_B_evt1.fits detnam echo+
HRC−S
```

The `instrume` value is `hrc−i` for file A and `hrc−s` for file B.

The header keyword `RANGELEV` must be present in order to apply the `AMP_SF` correction:

```
unix% dmkeypar hrc_evt1.fits RANGELEV echo+
# dmkeypar (CIAO 3.4): ERROR: Keyword 'RANGELEV' was not found in file 'hrc_evt1.fits'.
```

In this case, it is necessary to add the keyword; the value is determined by the `DATE−OBS` keyword:

```
unix% dmkeypar hrc_evt1.fits DATE−OBS echo+
2001−03−09T03:02:44
```

Now choose the appropriate `RANGELEV` value from this table:

`DATE−OBS`	Detector	`RANGELEV` value
before 1999−12−6	HRC−I & HRC−S	90

after 1999−12−6	HRC−I	115
after 1999−12−6	HRC−S	125

and add it to the header:

```
unix% dmhedit infile=hrc_evt1.fits filelist=none operation=add key=RANGELEV value=<value>
```

Now run the tool:

```
unix% punlearn hrc_process_events
unix% pset hrc_process_events infile=hrc_evt1.fits
unix% pset hrc_process_events outfile=hrc_new_evt1.fits
unix% pset hrc_process_events badpixfile=hrc_bpix1.fits
unix% pset hrc_process_events acaofffile=pcad_asol1.fits
unix% pset hrc_process_events instrume=hrc−s
unix% pset hrc_process_events do_amp_sf=yes
unix% pset hrc_process_events badfile=NONE
unix% hrc_process_events
input level 0 event file/stack (hrc_evt1.fits):
output level 1 file (hrc_new_evt1.fits):
bad pixel file ( NONE | none | <filename>) (hrc_bpix1.fits):
# hrc_process_events (CIAO 3.4): The following error occurred 2482 times:
        dsHPEEVENTSEQERR −− WARNING: Out of sequence events discovered in hrc_evt1.fits.
```

In some cases there will be more than one asol1.fits file for an observation. ***All*** the files must be input to the `acaofffile` parameter, either as a list or as a stack (see ahelp stack for more information).

The warning may be ignored; it is explained in this FAQ.

The contents of the parameter file may be checked using plist hrc_process_events.

Question 2: is your observation HRC−S imaging, HRC−I imaging, or grating data?

- HRC−S Imaging
- HRC−I Imaging
- Grating

HRC−S Imaging Observations

There are two filtering steps that need to be completed for HRC−S imaging observations:

1. Apply the status filter that is specific to HRC−S observations; a value of 0 demands that the bit be flagged as "good", a value of x indicates that either status (0/1) is acceptable. The pha filter removes about 25% of the background with no X−ray losses; see this FAQ:

   ```
   unix% punlearn dmcopy
   unix% dmcopy \
         "hrc_new_evt1.fits[pha=0:254,status=xxxxxx00xxxx0xxx0000x000x00000xx]" \
         hrc_flt_evt1.fits
   ```

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

```
unix% punlearn dmcopy
unix% dmcopy \
      "hrc_flt_evt1.fits[EVENTS][@hrc_std_flt1.fits][cols −crsu,−crsv,−amp_sf,−av1,−av2,−av3,
      hrc_evt2.fits
```

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

The thread is now complete; hrc_evt2.fits is a level=2 event file with the latest calibration products applied.

HRC−I Imaging Observations

There are two filtering steps that need to be completed for HRC−I imaging observations:

1. Apply the status filter that is specific to HRC−I observations; a value of 0 demands that the bit be flagged as "good", a value of x indicates that either status (0/1) is acceptable:

```
unix% punlearn dmcopy
unix% dmcopy \
      "hrc_new_evt1.fits[status=xxxxxx00xxxx0xxx00000000x0000000]"  \
      hrc_flt_evt1.fits
```

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

```
unix% punlearn dmcopy
unix% dmcopy \
      "hrc_flt_evt1.fits[EVENTS][@hrc_std_flt1.fits][cols −crsu,−crsv,−amp_sf,−av1,−av2,−av3,
      hrc_evt2.fits
```

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

The thread is now complete; hrc_evt2.fits is a level=2 event file with the latest calibration products applied.

Grating Observations

There are two CIAO threads that deal specifically with generating a level=2 event file for an HRC grating observation. Please choose the appropriate one, depending on which detector was used; note there is a separate thread for analysing multiple sources:

Single source:

- Obtain Grating Spectra from LETG/HRC−S Data
- Obtain Grating Spectra from LETG/HRC−I Data

Multiple sources:

- Obtain Grating Spectra for Multiple Sources − HRC (applies to both HRC−S and HRC−I)

Once you leave this thread, you ***do not*** need to return to complete any further steps.

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


#--------------------------------------------------------------------------
#
#  acis_process_events.par- Parameter file for acis_process_events program
#
#--------------------------------------------------------------------------
        infile = acis_evt1.fits   Input event file or stack
       outfile = acis_new_evt1.fits Output event file name
    acaofffile = pcad_asol1.fits  aspect offset file ( NONE | none | <filename>)
    (apply_cti = yes)             Apply CTI adjustment?
  (apply_tgain = yes)             Apply time-dependent gain adjustment?
(alignmentfile = )acaofffile -> pcad_asol1.fits) 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_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:pha,l:ph
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 = sky)             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,f:energ
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_pha,f:e
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,l:pha_r
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_ro,s:cor
f:energy,l:pi,s:fltgrade,s:grade,x:status}) cc graded event format definition string
         (mode = ql)
```

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


        infile = acis_evt2.fits   Input dataset/block specification
       outfile = acis_dstrk_evt2.fits Output dataset/block specification
          (max = )                streak threshold  syntax:  m  OR  m:m:m:m
(max_rowloss_fraction = 5.0e−5)          Maximum fraction of avg streaks/node/frame
    (num_sigma = 1.0)             Sigma value for determining streak threshold
       (filter = yes)             Discard tagged events
         (mask = [status=0,grade=0,2:4,6]) Filter to select candidate streak events
       (ccd_id = 8)               CCD ID to filter
      (ccd_col = ccd_id)          CCD ID column name
     (node_col = node_id)         Node ID column name ('none' for single node)
      (exptime = −1)              frame time (s) (reads EXPTIME if no pos. value given)
    (countfile = )                filename for event row−count distribution
     (fracfile = )                filename for cumulative streak contam function
     (timefile = )                filename for exposure time lost per row
      (verbose = 0)               Debug Level(0−5)
      (clobber = no)              Clobber existing file
         (mode = ql)
```

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


#
#  Parameters for the hrc_process_events task
#
        infile = hrc_evt1.fits    input level 0 event file/stack
       outfile = hrc_new_evt1.fits output level 1 file
    badpixfile = hrc_bpix1.fits   bad pixel file ( NONE | none | <filename>)
    acaofffile = pcad_asol1.fits  aspect offset file ( NONE | none | <filename>)
      (geompar = geom)            Parameter file for Pixlib Geometry files
(alignmentfile = )acaofffile −> pcad_asol1.fits) sim/fam alignment file ( NONE | none | <filename>)
      (obsfile = hrc_obs.par)     obs.par file for output file keywords ( NONE | none | <filename>)
     (gainfile = CALDB)           gain correction image file ( NONE | none | <filename>)
      (ADCfile = NONE)            ADC correction table file ( NONE | none | <filename>)
    (degapfile = CALDB)           degap factors (NONE | none | COEFF | <filename>)
      (hypfile = CALDB)           Hyperbolic test coefficients file ( NONE | none | <filename>)
 (ampsfcorfile = CALDB)           caldb file for amp_sf_correction( NONE | none | <filename>)
      (tapfile = CALDB)           tap ring test coefficients file ( NONE | none | <filename>)
   (ampsatfile = CALDB)           ADC saturation test file ( NONE | none | <filename>)
  (evtflatfile = CALDB)           Event flatness test file ( NONE | none | <filename>)
      (badfile = NONE)            output level 1 bad event file
      (logfile = stdout)          debug log file (STDOUT | stdout | <filename>)
     (eventdef = )stdlev1 −> {d:time,s:crsv,s:crsu,s:amp_sf,s:av1,s:av2,s:av3,s:au1,s:au2,s:au3,
l:raw,s:chip,l:tdet,f:det,f:sky,s:pha,s:pi,s:sumamps,s:chip_id,x:status}) output format definition
  (badeventdef = )badlev1 −> {d:time,s:crsu,s:crsv,s:au1,s:au2,s:au3,s:av1,s:av2,s:av3,s:pha}) outpu
   (grid_ratio = 0.5)             charge ratio
    (pha_ratio = 0.5)             pha ratio
  (wire_charge = 0)               turn on center wire test (−1=off,0=on)
         (cfu1 = 1.0)             u axis 1st order cor. factor
         (cfu2 = 0)               u axis 2nd order cor. factor
         (cfv1 = 1.0)             v axis 1st order cor. factor
         (cfv2 = 0)               v axis 2nd order cor. factor
  (time_offset = 0)               Offset to add to event time field to synch w/ fam data
     (amp_gain = 75.0)            amp gain
     (instrume = hrc−s)           hrc instrument− used for parameter file
(do_amp_sf_cor = yes)             perform amp_sf correction (y/n) ?
```

```
     (do_ratio = yes)             perform ratio validity checks
       (tstart = TSTART)           header key containing default time value (HSI)
        (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)
    (rand_seed = 1)               random seed (for pixlib), 0 = use time dependent seed
(rand_pix_size = 0.0)             pixel randomization width (−size..+size), 0.0 = no randomization
        (start = coarse)          start transformations at [coarse,chip,tdet]
         (stop = sky)             end transformations at [none,chip,tdet,det,sky]
      (badlev1 = {d:time,s:crsu,s:crsv,s:au1,s:au2,s:au3,s:av1,s:av2,s:av3,s:pha}) event format definition
      (hsilev1 = {d:time,s:crsu,s:crsv,s:au1,s:au2,s:au3,s:av1,s:av2,s:av3,s:chipx,s:chipy,s:tdetx,s:tdety,
s:y,l:fpz,s:pha,s:vstat,s:estat}) event format definition string
      (stdlev1 = {d:time,s:crsv,s:crsu,s:amp_sf,s:av1,s:av2,s:av3,s:au1,s:au2,s:au3,l:raw,s:chip,l:tdet,f:d
f:sky,s:pha,s:pi,s:sumamps,s:chip_id,x:status}) event format definition string
      (simlev1 = {l:tick,i:scifr,i:mjf,s:mnf,s:evtctr,s:crsu,s:crsv,s:au1,s:au2,s:au3,s:av1,s:av2,s:av3,
s:tdetx,s:tdety,s:pha,s:vstat,s:estat}) sim event definition string
      (fltlev1 = {d:time,s:crsv,s:crsu,s:amp_sf,s:av1,s:av2,s:av3,s:au1,s:au2,s:au3,s:chipx,s:chipy,l:tdetx
l:tdety,s:detx,s:dety,s:x,s:y,s:pha,s:sumamps,s:chip_id,l:status}) event format definition string
         (mode = ql)
```

History

22 Dec 2004	updated for CIAO 3.2: minor changes to parameter files; use ACIS bad pixel file (`badpixfile` parameter)
12 Jan 2005	grating sections now include links to multiple−source analysis threads
01 Feb 2005	added note about "Event island contains 1 or more bad pixels" warning to `acis_process_events` section
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: the CTI and time−dependent gain corrections may be applied to CC−mode data (see the Continuous Clocking Mode why topic for details); parameter file updates for `destreak`
01 Dec 2006	updated for CIAO 3.4: CIAO version in errors and warnings

URL: http://cxc.harvard.edu/ciao/threads/createL2/ Last modified: 1 Dec 2006