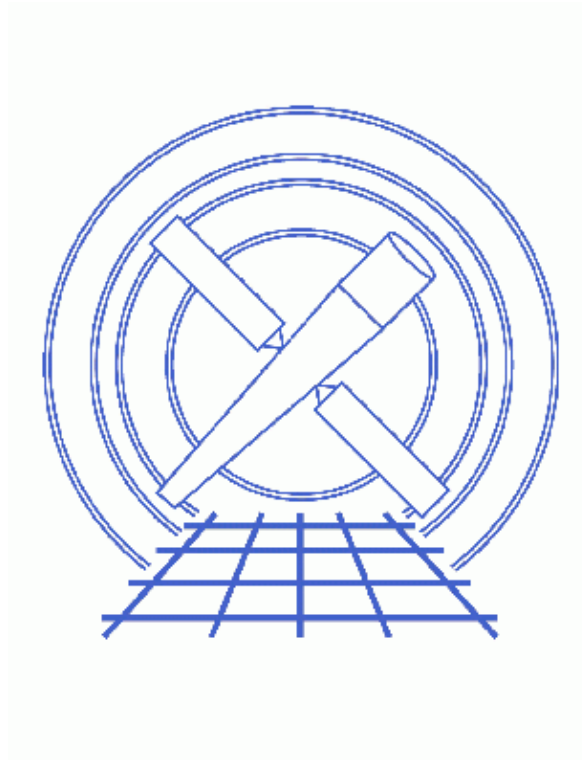


Reprocessing Data to Create a New Level=2 Event File



CIAO 4.1 Science Threads

Table of Contents

- **Do I have to run this thread?**
- **Get Started**
- **ACIS or HRC?**
- **ACIS Observations**
 - ◆ Imaging Observations
 - ◆ Grating Observations
 - ◆ Common acis_process_events Warnings
- **HRC Observations**
 - ◆ HRC-S Imaging Observations
 - ◆ HRC-I Imaging Observations
 - ◆ Grating Observations
 - ◆ Common hrc_process_events Warnings
- **History**

Reprocessing Data to Create a New Level=2 Event File

CIAO 4.1 Science Threads

Overview

Last Update: 7 Jan 2009 - updated for CIAO 4.1: `acis_process_events` no longer prints the "Event island contains 1 or more bad pixels." warning when `verbose=0`; other warnings were updated

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.**

Purpose:

To generate a new level=2 event file (`evt2.fits`) for all possible grating/detector combinations. This thread may be run on any Chandra observation.

If your data have gone through [Reprocessing III](#), you probably do not have to run this thread; the "[do I have to run this thread?](#)" section lists the exceptions and shows how to check the software version used in processing your data.

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.

Do I have to run this thread?

If your data have gone through [Reprocessing III](#), you most likely do not have to run this thread. Repro III started with DS 7.6.7.

To find the software version that was used on the data, check the ASCDSVER keyword in the header:

```
unix% dmkeypar acis_evt1.fits ASCDSVER echo+  
6.5.1
```

Create a New Level=2 Event File - CIAO 4.1

- If the ASCDSVER is lower than DS 7.6.7: **yes**, you should run this thread.
- If the ASCDSVER is DS 7.6.7 or higher: **no**, you probably do not have to run this thread.

Note that the version numbering changed after version R4CU5UPD14 to the "DS" system, starting with DS 6.0.0.

Exceptions to the version check

- Newest ACIS TGAIN calibration (-120 C data)
- ACIS TGAIN calibration for -110 C data
- ACIS CTI calibration for ACIS-S1 and S3
- VFAINT mode data

Newest ACIS TGAIN calibration (-120 C data)

The ACIS time-dependent gain (TGAIN) calibration files for -120 C focal plane temperature are frequently updated in the calibration database (CALDB). If your data were taken recently - even if the ASCDSVER is DS 7.6.7 or higher - newer TGAIN file may be available for your dataset.

The Improvements in Calibration section of the TGAIN why topic lists the updates by date and CALDB version. The observation date and focal plane temperature are recorded in the header of the event file:

```
unix% dmkeypar acis_evt1.fits DATE-OBS echo+
2008-01-19T20:41:39

unix% dmkeypar acis_evt1.fits FP_TEMP echo+
153.60722351
```

If there is a new TGAIN calibration for your data, run this thread to apply the new calibration.

ACIS TGAIN calibration for -110 C data

ACIS time-dependent gain (TGAIN) calibration files for -110 C focal plane temperature were released for the first time in CALDB 3.4.3 (31 March 2008).

This calibration applies only to -110 C data taken on the back-illuminated chips S1 (ACIS-5) or S3 (ACIS-7). Note that there is not a CTI correction available for this data, only TGAIN.

The focal plane temperature and detector name are recorded in the header of the event file:

```
unix% dmkeypar acis_evt1.fits FP_TEMP echo+
163.952042

unix% dmkeypar acis_evt1.fits DETNAM echo+
ACIS-456789
```

Run this thread to apply the TGAIN correction to chips S1 and S3. The remaining front-illuminated (FI) chips in the file will be reprocessed as usual during the run.

ACIS CTI calibration for ACIS-S1 and S3

The CTI calibration (both parallel and serial) for the back-illuminated chips (ACIS-S1, S3) was released in CALDB v3.3.0 (18 December 2006). The majority of Repro III was already done by December 2006, so this calibration was not applied to all data.

This calibration applies to -120 C data taken on the back-illuminated chips S1 (ACIS-5) or S3 (ACIS-7).

The focal plane temperature, detector name, and CALDB version are recorded in the header of the event file:

```
████████████████████████████████████████████████████████████████████████████████
```

Create a New Level=2 Event File - CIAO 4.1

```
unix% dmkeypar acis_evt1.fits FP_TEMP echo+
153.952042

unix% dmkeypar acis_evt1.fits DETNAM echo+
ACIS-456789

unix% dmkeypar acis_evt1.fits CALDBVER echo+
3.2.1
```

For this file, S1 and S3 were used during the observation, but the CALDB version is lower than CALDB 3.3.0. Run this thread to apply the CTI correction to chips S1 and S3.

VFAINT mode data

The very faint (VFAINT) mode background cleaning **is not** included in this thread, since there are several details to be considered. The data mode is recorded in the DATAMODE header keyword:

```
unix% dmkeypar acis_evt1.fits DATAMODE echo+
VFAINT
```

Users with VFAINT data should follow the [Clean ACIS Background in VFAINT Mode thread](#) instead of this thread.

[Proceed to the next section](#) to begin.

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.](#)

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

Running `acis_process_events` produces a new level=1 event file that has the latest calibration applied:

- [charge transfer inefficiency \(CTI\) correction](#)
- [time-dependent gain adjustment](#)
- [gain map](#)
- [PHA randomization](#)

Create a New Level=2 Event File - CIAO 4.1

- [pixel randomization](#)
- [continuous clocking mode times of arrival](#) (be sure to read the [data caveats](#))

Determine the eventdef

The `eventdef` parameter specifies the names and data types of the columns in the output event data file. Four definitions 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

The event mode of an observation 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."

Set the parameters

If you created a new bad pixel file by running the [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.

All of the calibration options are "on" by default in `acis_process_events`, but are `pset` for emphasis.

```
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.lis
unix% pset acis_process_events eventdef=")stdlev1"
unix% pset acis_process_events rand_pix_size=0.5
unix% pset acis_process_events rand_pha=yes
unix% pset acis_process_events apply_tgain=yes
unix% pset acis_process_events apply_cti=yes
```

It is important to note several things here:

- In some cases, there will be more than one aspect solution file (`pcad_asol1.fits`) for an observation. **All** the files must be input to the `acaofffile` parameter **in chronological order**, either as a comma-separated list or **as a stack**. For example:

```
unix% cat pcad_asol1.lis
pcadf063874624N002_asol1.fits
pcadf063875522N002_asol1.fits
pcadf063902942N002_asol1.fits
```

The time is in the filename, so "ls" lists them in chronological order.

Create a New Level=2 Event File - CIAO 4.1

- 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. A warning will be printed to the screen.
- To calculate the continuous clocking mode times of arrival, set:

```
unix% pset acis_process_events calc_cc_times=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_asoll1.fits):
```

A number of possible warning messages are explained in the [Common acis_process_events Warnings section](#).

Question 2: is your observation imaging or grating data?

- [Imaging](#)
 - [Grating](#)
-

Imaging Observations

There are two filtering steps for ACIS imaging observations:

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

```
unix% punlearn dmcop
unix% dmcop "acis_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
  acisflt_evt1.fits
```

2. The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied; these are stored in the `acisflt1.fits` file. Simultaneously, unnecessary columns are eliminated from the output:

```
unix% punlearn dmcop
unix% dmcop "acisflt_evt1.fits[EVENTS][@acisflt1.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. For details on how the tool works, see the [Destreaking ACIS Data](#) why topic and "[ahelp destreak](#)".

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 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. Follow the appropriate one, depending on which transmission grating was used:

Single source:

- [HETG/ACIS-S Grating Spectra](#)
- [LETG/ACIS-S Grating Spectra](#)

Multiple sources:

- [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.

Common `acis_process_events` Warnings

It is important to remember that the following messages are **warnings, not errors**. The tool may print one of these messages, but will still create a valid event file.

WARNING: problem reading ctifile, cti adjustment will not be applied.

```
# acis_process_events (CIAO 4.1): WARNING: problem reading ctifile, cti adjustment will not be applied
```

This warning may also be triggered by `product=T_GAIN`.

The most common cause for seeing this warning is that the observation does not contain -120C data. Since the CTI and TGAIN is only calibrated for this focal place temperature, it is not possible to use it on other observations.

Since no file is found, `acis_process_events` continues running as if `apply_cti=no` or `apply_tgain=no` and produces a valid output file that has not been CTI-corrected or had the TGAIN applied.

The CTI correction cannot be performed on a graded mode event list.

```
# acis_process_events (CIAO 4.1): WARNING: The CTI correction cannot be performed on a graded mode event list
# acis_process_events (CIAO 4.1): WARNING: Although the keyword TGAINCOR is false or missing, the
# acis_process_events (CIAO 4.1): WARNING: The values in this column are assumed to be the unadjusted pulse
```

The CTI correction requires that a PHAS column exist in the input file. This column is not created for observations taken with the data mode GRADED, so this warning is produced.

In this case, `acis_process_events` continues to reprocess the data as if `apply_cti=no` and produces a valid output file that has not been CTI-corrected.

HRC Observations

Running `hrc_process_events` produces a new level=1 event file that has the latest calibration applied:

- [AMP SF correction](#)
- [degap correction](#)

Determine the "instrume" parameter

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.

Determine the RANGELEV value

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

```
unix% dmkeypar hrc_evt1.fits RANGELEV echo+
# dmkeypar (CIAO 4.1): 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>
```

Set the parameters

```
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.lis
unix% pset hrc_process_events badfile=NONE
unix% pset hrc_process_events instrume=hrc-s
unix% pset hrc_process_events do_amp_sf_cor=yes
```

- In some cases, there will be more than one aspect solution file (`pcad_asol1.fits`) for an observation. **All** the files must be input to the `acaofffile` parameter **in chronological order**, either as a comma-separated list or **as a stack**. For example:

```
unix% cat pcad_asol1.lis
pcadf062419415N003_asol1.fits
pcadf062461940N003_asol1.fits
pcadf062492182N003_asol1.fits
```

The time is in the filename, so "ls" lists them in chronological order.

- Set the `instrume` parameter (here, `hrc-s`) to the correct value for the dataset.

Execute the tool

```
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):
```

A number of possible warning messages are explained in the [Common hrc_process_events Warnings section](#).

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 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]" \
  hrcflt_evt1.fits
```

2. The [Good Time Intervals](#) (GTIs) supplied by the pipeline now need to be applied; these are stored in the hrc_stdflt1.fits file. Simultaneously, unnecessary columns are eliminated from the output:

```
unix% punlearn dmcopy
unix% dmcopy \
  "hrcflt_evt1.fits[EVENTS][@hrc_stdflt1.fits][cols -crsu,-crsv,-amp_sf,-av1,-av2,-av3,-au"
  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.

Next, read the Computing Average HRC Dead Time Corrections thread ([S-Lang](#) or [Python](#)) to see if you need to recompute the deadtime statistics for your dataset.

HRC-I Imaging Observations

There are two filtering steps 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]" \
  hrcflt_evt1.fits
```

2. The [Good Time Intervals](#) (GTIs) supplied by the pipeline now need to be applied; these are stored in the hrc_stdflt1.fits file. Simultaneously, unnecessary columns are eliminated from the output:

Create a New Level=2 Event File - CIAO 4.1

```
unix% punlearn dmcop
unix% dmcop \
    "hrcflt_evt1.fits[EVENTS] [@hrc_stdflt1.fits][cols -crsu,-crsv,-amp_sf,-av1,-av2,-a
    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.

Next, read the Computing Average HRC Dead Time Corrections thread ([S-Lang](#) or [Python](#)) to see if you need to recompute the deadtime statistics for your dataset.

Grating Observations

There are two CIAO threads that deal specifically with generating a level=2 event file for an HRC grating observation. Follow the appropriate one, depending on which detector was used:

Single source:

- [LETG/HRC-S Grating Spectra](#)
- [LETG/HRC-I Grating Spectra](#)

Multiple sources:

- [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.

Common hrc_process_events Warnings

It is important to remember that the following messages are **warnings, not errors**. The tool may print one of these messages, but will still create a valid event file.

Out of sequence events discovered in hrc_evt1.fits.

```
# hrc_process_events (CIAO 4.1): The following error occurred 2482 times:
    dsHPPEVENTSEQERR -- WARNING: Out of sequence events discovered in hrc_evt1.fits.
```

There are occasional events that appear in the telemetry stream with time tags that make them seem to have occurred out-of-sequence. One special case where this occurs has been documented (see the [Out-of-sequence HRC Time Tags](#) memo); other occurrences are most likely to be caused by hiccups in the HRC or by double-bit errors in telemetry (single-bit errors are corrected).

This warning may be ignored in most cases, as long as the number of events flagged as out-of-sequence is a small fraction of the total number of events in the file; see [this FAQ](#) for more information.

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

Create a New Level=2 Event File - CIAO 4.1

- 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
- 08 Feb 2008 updated for CIAO 4.0: links point to why topics and dictionary entries instead of other ACIS and HRC threads, which have been deprecated
- 31 Mar 2008 updated for CALDB 3.4.3: new -120 C TGAIN files and first -110 C TGAIN file: both changes made in [Do I have to run this thread? section](#)
- 16 Jun 2008 added CTI correction for back-illuminated chips to [Do I have to run this thread? section](#)
- 07 Jan 2009 updated for CIAO 4.1: `acis_process_events` no longer prints the "Event island contains 1 or more bad pixels." warning when `verbose=0`; other warnings were updated

URL: <http://cxc.harvard.edu/ciao/threads/createL2/>

Last modified: 7 Jan 2009