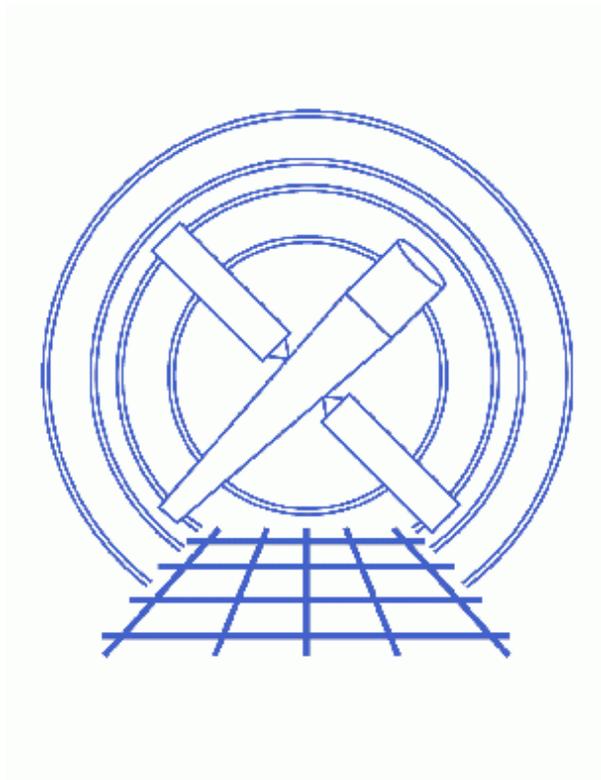


# Clean ACIS Background in VFAINT Mode



## ***CIAO 3.4 Science Threads***

# Table of Contents

- ***Background Information***
- ***Get Started***
  - ◆ Related acis\_process\_events threads
- ***Generate A New Level=1 Event File***
  - ◆ Determine the eventdef parameter
  - ◆ Run acis\_process\_events
- ***Generate A New Level=2 Event File***
  - ◆ Apply grade/status filters
  - ◆ Apply GTI filters
- ***Caveats***
  - ◆ Elimination of real events
  - ◆ Combining with destreak
  - ◆ Combining with ACIS CTI Correction
- ***Parameter files:***
  - ◆ acis\_process\_events
- ***History***
- ***Images***
  - ◆ Image of "Cleaned" Photons

# Clean ACIS Background in VFAINT Mode

*CIAO 3.4 Science Threads*

## Overview

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

### **Synopsis:**

In ACIS very faint mode, there is a 5x5 pixel event island, instead of just a 3x3 event island (as in FAINT mode). Therefore, `acis_process_events` can use the pulse heights in the outer 16 pixels of the 5x5 event island to help distinguish between good X-ray events and bad events that are most likely associated with cosmic rays. The [Background Information](#) section has more details.

### **Purpose:**

To clean the ACIS particle background for very faint mode observations.

### **Read this thread if:**

you are working with an ACIS observation that was taken in VFAINT mode; [Get Started](#) shows how to check the mode of your observation.

### **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. Note that the CTI correction is on by default (`apply_cti=yes`) in `acis_process_events`.
- `clean55` tool: [contributed software](#) that performs the same function as this thread.

**Proceed to the [HTML](#) or [hardcopy \(PDF: A4 | letter\)](#) version of the thread.**

---

## Background Information

In VFAINT mode, this is the benefit of working with a 5x5 pixel event island, instead of just a 3x3 event island (as in FAINT mode). Therefore, `acis_process_events` can use the pulse heights in the outer 16 pixels of the 5x5 event island to help distinguish between good X-ray events and bad events that are most likely associated with cosmic rays. An event is considered to be a potential background event if one or more of those outer pixels is greater than a set split threshold. In that case, status bit 23 is set to one ("bad") for the event.

## Clean ACIS Background in VFAINT Mode – CIAO 3.4

More information on this is available from the [Reducing ACIS Quiescent Background Using Very Faint Mode](#) page. The abstract from that page is included here:

ACIS particle background can be reduced significantly compared to the standard grade selection by screening out events with significant flux in border pixels of the 5x5 event islands. The particle background above 6 keV is reduced by a factor of 1.4 in the front-illuminated chips and ~ 1.25 in the back-illuminated chips. The background rejection is much better at soft energies – by a factor of 2 near 0.5 keV in FI chips and by a factor of 3 near 0.3 in BI chips. In the intermediate energies, 1–5 keV, the background is reduced by a factor of 1.1–1.15.

Please be sure to read the [Caveats](#), especially if you are dealing with a bright source.

---

## Get Started

*Sample ObsID used:* 884 (ACIS–S, 0235+164)

*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.

## 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](#)
- [Apply an ACIS Gain Map](#)
- [Remove Pixel Randomization](#)
- [Apply/Remove PHA Randomization](#)
- [Apply the ACIS CTI Correction](#)

---

## Generate A New Level=1 Event File

### Determine the eventdef parameter

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 acisf00884_000N001_evt1.fits READMODE echo+
TIMED

unix% dmkeypar acisf00884_000N001_evt1.fits DATAMODE echo+
VFAINT
```

This is indeed a very faint mode observation; all VFAINT observations will also have READMODE equal to

TIMED. The proper `eventdef` parameter, used in the [next section](#), is "stdlev1". The full parameter syntax of the `eventdef` string may be found in [plist acis\\_process\\_events](#).

---

## Run `acis_process_events`

Running `acis_process_events` with the `SDP level=1` event file as the input and with `check_vf_pha=yes` will produce a *new* `level=1` event file with possible background events flagged in status bit 23.

At the same time, the latest CALDB will be applied, meaning that the newest gain map will be picked up. The [ACIS CTI correction](#) is also applied while cleaning the background. This may not be appropriate for your source; please make sure to read the [Combining with ACIS CTI Correction caveat](#).

```
unix% punlearn acis_process_events
unix% pset acis_process_events infile=acisf00884_000N001_evt1.fits
unix% pset acis_process_events outfile=acis_884_new_evt1.fits
unix% pset acis_process_events badpixfile=acis_884_new_bpix1.fits
unix% pset acis_process_events eventdef=")stdlev1"
unix% pset acis_process_events stop=none
unix% pset acis_process_events check_vf_pha=yes
unix% acis_process_events
Input event file or stack (acisf00884_000N001_evt1.fits):
Output event file name (acis_884_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.

The `trail` parameter is also used when `check_vf_pha=yes`. The idea is that for the pixels above the center of the event island, the amount of charge in these pixels is due in part to a fraction of the charge that is trailed from the other pixels as a result of [charge transfer inefficiency \(CTI\)](#). In this case, the charge due to the CTI trail should be ignored when checking to see if the pixel is above the split threshold. **By default the algorithm ignores 2.7% of the charge in the central pixels.** This value comes from empirical data taken at the focal plane temperature of  $-110\text{C}$ . Those experiments have shown very little impact of `trail` parameter on the particle rejection efficiency. This is likely due to the fact that the CTI coefficient only affects testing of 1 or 2 of the 16 border pixels. Even if the value is far off, the rejection efficiency is degraded by 10% at most – that is, you reject 18% (instead of 20%) of the particles.

---

## 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 (i.e. all bits set to 0). Since the background particles are flagged as one ("bad"), they will be filtered out of the event file:

```
unix% punlearn dmcop
unix% dmcop "acis_884_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
  acis_884flt_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 dmcop
unix% dmcop \
  "acis_884flt_evt1.fits[EVENTS][@acisf00884_000N001flt1.fits][cols -phas]" \
  acis_884_evt2.fits
```

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

## Caveats

### Elimination of real events

*This caveat applies to both imaging and grating observations:* in bright sources where pile-up superimposes events on one another, some percentage of real X-ray events are marked as potential background events.

To see if this is a problem in your data, create a file of the photons rejected by the `check_vf_pha` screening. Here the filter is defined with status bit 23 marked as "bad" (1) and all other bits may be good or bad (x); note that the bits are counted from the right, starting at zero. This filter retains those events that have been flagged on account of the 5x5 screening:

```
unix% dmcop "acis_884_new_evt1.fits[EVENTS][status=xxxxxxxx1xxxxxxxxxxxxxxxxxxxxxxxx]" \
  acis_884_23bad.fits
```

Figure 1  shows `acis_884_new_evt1.fits` (left) and `acis_884_23bad.fits` (right) displayed side-by-side. As you can see, there are some source photons identified as background particles by `acis_process_events`; this is evident because the photons are clumped around the source location. In this case, you may decide to keep the photons (i.e. not use the newly screened file).

If the rejected photons are uniform across the field, then pile-up was not a problem and it is perfectly safe to filter out those events. There are more examples of this comparison on the [Reducing ACIS Quiescent Background Using Very Faint Mode](#) page.

## Combining with destreak

When the `destreak` tool is run on data that has been processed with `check_vf_pha=yes`, it is not as efficient at detecting (and removing) streak events because of the large number of nonzero status bits set by the background-cleaning algorithm. While the `destreak` [mask parameter](#) may now be used to include the nonzero status events in the destreaking, the following is still the recommended method for combining these two processes:

1. run `destreak` on the level=1 event file, as shown in the [Destreak the ACIS-S4 Chip](#) thread
2. perform the VFAINT background cleaning with `acis_process_events` (i.e. run this thread)

## Combining with ACIS CTI Correction

It has been determined that for faint sources the number of events flagged by the VFAINT algorithm is insensitive to the use of the CTI adjustment. As a result, the VFAINT background algorithm and the CTI adjustment may be used in the same run of `acis_process_events`, as shown in this thread.

If the source is not faint, it is suspected that simultaneous use of the VFAINT background algorithm and the CTI adjustment may flag more source events as background. In this case, one may wish to apply the VF mode filter first, then rerun `acis_process_events` with the CTI adjustment. To do so:

1. Run `acis_process_events` as written in this thread, but turn off the CTI correction *and* the time-dependent gain correction:

```
unix% pset acis_process_events apply_cti=no
unix% pset acis_process_events doevtgrade=no
unix% pset acis_process_events apply_tgain=no
```

It is not necessary to recalculate the event grading (`doevtgrade=no`). Since the `tgain` correction requires CTI to be applied at the same time, it is turned off to avoid error messages.

2. Re-run `acis_process_events` to apply the CTI, as shown in the [Apply the ACIS CTI Correction thread](#).

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

```
#-----
#
# acis_process_events.par- Parameter file for acis_process_events program
#
#-----
      infile = acisf00884_000N001_evt1.fits   Input event file or stack
      outfile = acis_884_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>)
```

## Clean ACIS Background in VFAINT Mode – CIAO 3.4

```
(gradefile = CALDB)           grade mapping file ( NONE | none | CALDB | <filename>)
(gainfile = CALDB)           acis gain file ( NONE | none | CALDB | <filename>)
(badpixfile = acis_884_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:pi
l:pi,s:fltgrade,s:grade,x:status}) output format definition
(doevtgrade = yes)          Determine event flight grade?
(check_vf_pha = yes)        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)              where to end transformations
(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:energy
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:energy,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_ld,s:phas,l:pha,l:pha_ro,
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_ld,l:pha,l:pha_ro,s:corn_pha,
f:energy,l:pi,s:fltgrade,s:grade,x:status}) cc graded event format definition string
(mode = ql)
```

---

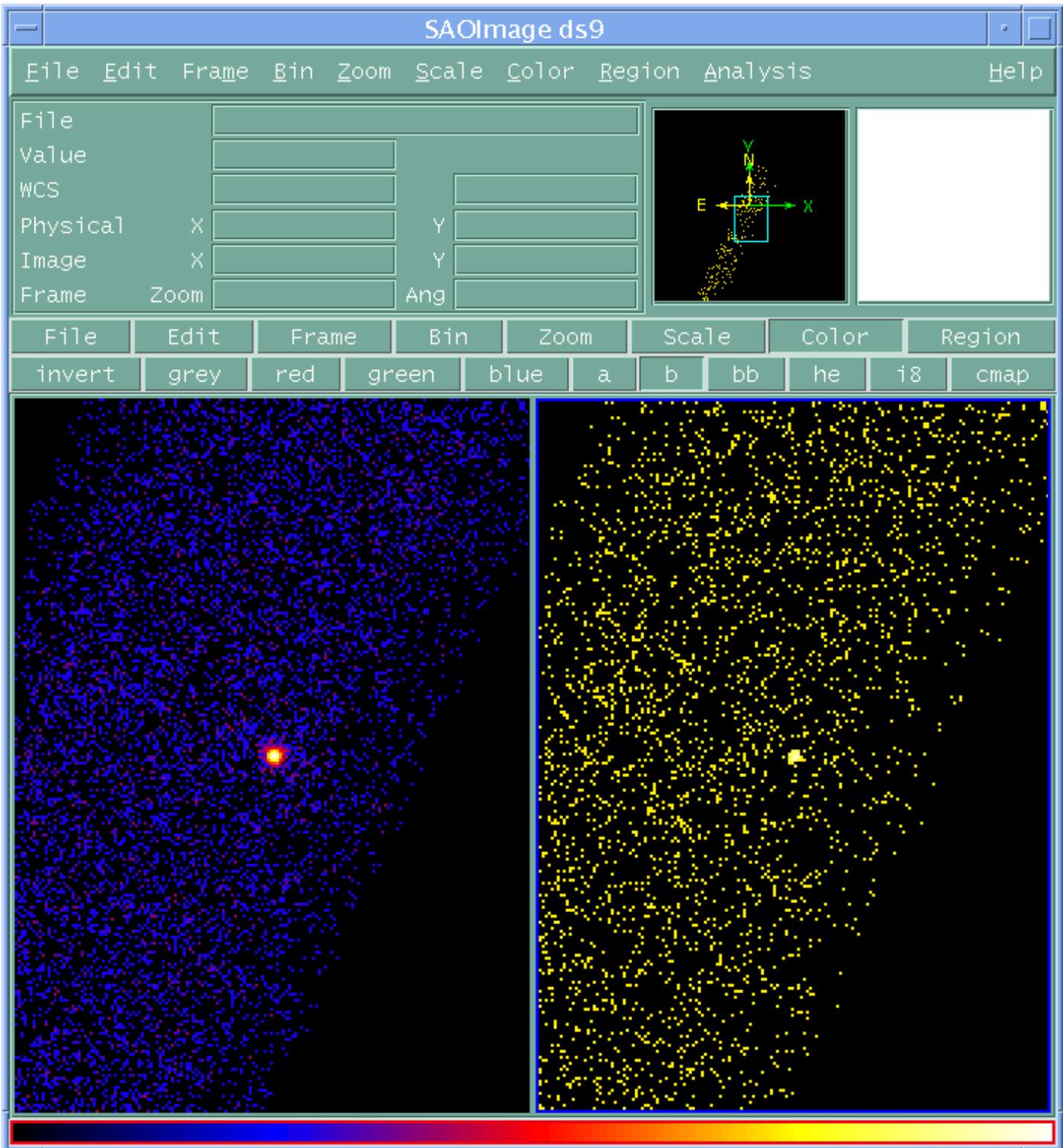
## History

- 16 Dec 2004 updated for CIAO 3.2: use ACIS bad pixel file (badpixfile parameter)
- 01 Feb 2005 added note about "Event island contains 1 or more bad pixels" warning
- 09 Feb 2005 added [Combining with destreak caveat](#)
- 20 Jun 2005 CIAO 3.2.2 patch: minor acis\_process\_events parameter change (default value of [threshfile](#) is CALDB instead of NONE)
- 09 Dec 2005 updated for CIAO 3.3: output filenames include ObsID
- 01 Dec 2006 updated for CIAO 3.4: 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](#))
- 21 May 2007 need to set [stop=none](#) if aspect solution is not provided

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

Last modified: 21 May 2007

### Image 1: Image of "Cleaned" Photons



The new level=1 event file, `acis_884_new_evt1.fits`, is displayed on the left and the file of photons marked as background particles, `acis_884_23bad.fits`, is on the right.