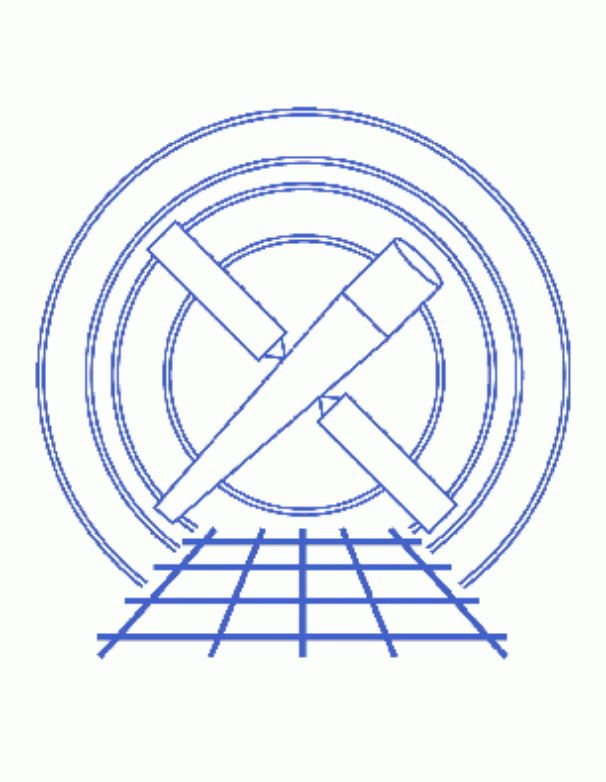
Calculate CC−mode Times of Arrival



CIAO 3.4 Science Threads

Table of Contents

- ***Get Started***
  - ♦ Caveat: TIMEDEL keyword
- ***Check Source Coordinates***
- ***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
- ***Summary***
- ***Parameter files:***
  - ♦ acis_process_events
- ***History***

Calculate CC−mode Times of Arrival

CIAO 3.4 Science Threads

Overview

Last Update: 18 Jun 2007 − expanded information in Check Source Coordinates section; included software version in which TIMEDEL keyword problem was corrected

Synopsis:

In continuous clocking mode event data files, the values recorded in the column `TIME` are the readout times instead of the times of arrival. (Note that the `TIME` values in timed exposure event data files are the times of arrival.) Before performing timing analyses, the times of arrival at the barycenter of the solar system should be computed. The first step of this process is to compute the times of arrival at the spacecraft from the readout times by removing the average time lag between the readout times and the times of arrival, the effect of the dither of the telescope, and the effect of motion of the detector relative to the telescope.

This correction became part of standard data processing (SDP) in DS 7.6.

Purpose:

To correct the event times in ACIS CC−mode observations for the time of arrival.

Read this thread if:

you are working with any ACIS CC−mode data observation, regardless of whether it is imaging or grating data processed with a software version lower than DS 7.6. The Get Started section shows how to find the software version used in SDP, and the Determine the eventdef parameter section shows how to find the mode of your observation.

Related Links:

- Why topic: Continuous Clocking Mode
- Apply Barycenter Correction thread: correct for the difference in photon arrival times as the Earth and Chandra move around the Sun. The barycenter correction should be done after running this thread.

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

Get Started

Sample ObsID used: 170 (ACIS−S/HETG, Crab Pulsar)

File types needed: evt1; flt1; asol1

In this thread, we assume that all relevant files are in the same working directory.

First, check the ASCDSVER header keyword to see if the correction was applied in SDP:

```
unix% dmkeypar acisf00170_001N003_evt1.fits ASCDSVER echo+
6.5.1
```

You can also determine if the times of arrival have already been computed for the observation (i.e. this thread has already been run) by checking for the HDUCLAS3 header keyword:

```
unix% dmkeypar acisf00170_001N003_evt1.fits HDUCLAS3 echo+
# dmkeypar (CIAO 3.4): ERROR: Keyword 'HDUCLAS3' was not found in file 'acisf00170_001N003_evt1.fits'.
```

If the correction had been applied, the HDUCLAS3 would have a value of CC_CORRECTED.

This observation was processed before the correction became part of SDP, i.e. the ASCDSVER is lower than DS 7.6. Also, the time−of−arrival correction has not previously been applied via acis_process_events. Therefore, this thread should be run.

Caveat: TIMEDEL keyword

The value of the keyword TIMEDEL in some ACIS continuous−clocking mode event−data files is incorrectly set to 1.4592 s instead of 0.00285 s. While this does not represent a problem if the pipeline−produced files are used for timing analyses, it is a problem if an affected event−data file is reprocessed using acis_process_events with the parameter calc_cc_times=yes because the values of the TIMEs in the output file will be corrupted.

This problem affects data processed between DS 7.6.0 (June 2005) and DS 7.6.8 (June 2006).

The dataset used in this thread − ObsID 170 − is not affected by the TIMEDEL problem:

```
unix% dmkeypar acisf00170_001N003_evt1.fits ASCDSVER echo+
6.5.1

unix% dmkeypar acisf00170_001N003_evt1.fits TIMEDEL echo+
0.00285
```

To see if your data file is affected, check the ASCDSVER (software version) and/or TIMEDEL header keywords in the evt1.fits file:

```
unix% dmkeypar acis_evt1.fits ASCDSVER echo+
7.6.2

unix% dmkeypar acis_evt1.fits TIMEDEL echo+
1.4592
```

This file has the incorrect TIMEDEL value, so dmhedit is used to update it:

```
unix% dmhedit infile=acis_evt1.fits filelist="" op=add key=TIMEDEL value=0.00285 unit=s
```

```
unix% dmkeypar acis_evt1.fits TIMEDEL echo+
0.00285
```

Check Source Coordinates

It is important to verify that the coordinates of the source are precise to less than 0.5 arcsec (i.e. one pixel). The source coordinates are used to determine the absolute times of arrival; if they are imprecise, the output times of arrival will also be inaccurate.

The coordinates that should be used are the best−known right ascension (RA_TARG) and declination (DEC_TARG) coordinates for the source:

```
unix% dmlist acisf00170_001N003_evt1.fits header | egrep '(RA|DEC)_TARG'
0057 RA_TARG                    83.6333330              Real8        Observer's specified target RA
0058 DEC_TARG                   22.0144720              Real8        Observer's specified target De
```

These coordinates in this file have the correct level of precision for the time−of−arrival correction.

If the accuracy of the coordinates in the header needs to be improved, here are some suggestions on how to determine the values:

- use previously−published source coordinate values.
- if the coordinates are not well known, try obtaining them from a Chandra observation of the source which was taken in timed−exposure mode.
- if neither of the above methods are available, it may be possible to use two, separate continuous−clocking mode observations to find the source coordinates, provided that the observations did not have the same roll angle.

It is not possible to fix the coordinates in the dataset if the position is not known from some other dataset or publication (i.e. you cannot determine accurate source coordinates from the dataset you are attempting to correct).

For reference, the conversion between arcsec and degrees is:

```
0.5 arcsec −> 0.00014 deg (in declination)
              0.00014 / cos(DEC_TARG) deg (in right ascension)
```

Once the new source coordinates are determined, modify the file header with dmhedit:

```
unix% dmhedit infile=acisf00170_001N003_evt1.fits filelist="" operation=add key=RA_TARG value='newva
```

Note that the uncertainties in the times of arrival include the uncertainties in the coordinates RA_TARG, DEC_TARG, the PSF, and the aspect reconstruction. If RA_TARG and DEC_TARG are accurate to much less than a pixel, the times of arrival should be accurate to about 4 ms for most observations (i.e. those without any unusual aspect problems).

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. Two *relevant* predefined strings are included in the parameter file for `acis_process_events`:

READMODE	DATAMODE	event mode	eventdef string
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 acisf00170_001N003_evt1.fits READMODE echo+
CONTINUOUS

unix% dmkeypar acisf00170_001N003_evt1.fits DATAMODE echo+
CC33_GRADED
```

This is a continuous clocking 3x3 graded observation, so the proper `eventdef` parameter is "ccgrdlev1".

After the times of arrival are calculated, they are stored in the `TIME` column of the event file and the previous `TIME` values (i.e. the readout time) are moved into a new column named `TIME_RO`. In order to retain this information in the output file, the eventdef string must be customized by adding "d:time_ro". Here we have taken the full text of "ccgrdlev1" (from the acis_process_events parameter file) and added the new column definition:

```
eventdef="{d:time,d:time_ro,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
```

The position of the column in the eventdef parameter, and therefore in the output file, is irrelevant to the CIAO tools. The *only* effect of doing this is that the `TIME_RO` column will be written to the output.

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 time of arrival calculated for each event. The aspect solution is supplied so that the dither and the SIM motion can be removed from the readout times.

In this example, it is not necessary to recalculate the grades (`doevtgrade=no`) or the energy and PI values (`calculate_pi=no`).

```
unix% punlearn acis_process_events
unix% pset acis_process_events infile=acisf00170_001N003_evt1.fits
unix% pset acis_process_events outfile=acis_170_new_evt1.fits
unix% pset acis_process_events acaofffile="@pcad_asol1.lis"
unix% pset acis_process_events eventdef="{d:time,d:time_ro,s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,
unix% pset acis_process_events doevtgrade=no
unix% pset acis_process_events calc_cc_times=yes
unix% pset acis_process_events calculate_pi=no
unix% pset acis_process_events stop=sky
unix% acis_process_events
Input event file or stack (acisf00170_001N003_evt1.fits):
Output event file name (acis_170_new_evt1.fits):
aspect offset file ( NONE | none | <filename>) (@pcad_asol1.lis):
```

```
# acis_process_events (CIAO 3.4): WARNING: The CTI correction cannot be performed on a graded mode e
# acis_process_events (CIAO 3.4): WARNING: No files found matching CALDB search:
        tel=CHANDRA
        inst=ACIS
        det=−
        filt=−
        product=T_GAIN
        start_date=1999−08−29T10:19:18
        start_time=10:19:18
        stop_date=1999−08−29T12:31:47
        stop_time=12:31:47
        query=CTI_APP.eq.NNNNNNNNNN

# acis_process_events (CIAO 3.4): WARNING: problem reading tgainfile, tgain adjustment will not be a
# acis_process_events (CIAO 3.4): WARNING: The values of ENERGY and PI may be inaccurate because cal
```

It is important to note several things here:

- In some cases there will be more than one asol1.fits file for an observation. ***All*** the files must a 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). For example, here we used:

  ```
  unix% more pcad_asol1.lis
  pcadf052309704N002_asol1.fits
  pcadf052317071N002_asol1.fits
  ```

- The warnings are generated because the CTI correction cannot be applied to `GRADED` mode data; see the Apply the ACIS CTI Correction thread for details. The warnings are harmless, but may be eliminated by setting:

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

  Note that these corrections ***should be applied*** to `FAINT` mode data (i.e. leave the parameters at the default value "yes").

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

Generate A New Level=2 Event File

Apply grade/status filters

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_170_new_evt1.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
      acis_170_flt_evt1.fits
```

Apply GTI filters

The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied:

```
unix% punlearn dmcopy
unix% dmcopy "acis_170_flt_evt1.fits[EVENTS][@acisf00170_001N003_flt1.fits]" \
      acis_170_evt2.fits
```

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

Summary

The thread is now complete; the new level=2 event file is `acis_170_evt2.fits`. To confirm that the times of arrival were calculated, check the header:

```
unix% dmkeypar acis_170_evt2.fits HDUCLAS3 echo+
CC_CORRECTED
```

Before beginning any timing analysis, complete the Apply Barycenter Correction thread to correct for the difference in photon arrival times as the Earth and Chandra move around the Sun.

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


#-------------------------------------------------------------------------
#
#  acis_process_events.par- Parameter file for acis_process_events program
#
#-------------------------------------------------------------------------
        infile = acisf00170_001N003_evt1.fits   Input event file or stack
       outfile = acis_170_new_evt1.fits Output event file name
    acaofffile = @pcad_asol1.lis  aspect offset file ( NONE | none | <filename>)
    (apply_cti = yes)             Apply CTI adjustment?
  (apply_tgain = yes)             Apply time-dependent gain adjustment?
(alignmentfile = )acaofffile -> @pcad_asol1.lis) 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 = NONE)            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 = {d:time,d:time_ro,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:corn_pha,f:energy,l:pi,s:fltgrade,s:grade,x:status}) output format definition
   (doevtgrade = no)              Determine event flight grade?
 (check_vf_pha = no)              Check very faint pixels?
```

```
(calc_cc_times = yes)            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 = no)             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,
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_
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

16 Dec 2004	reviewed for CIAO 3.2: no changes
09 May 2005	updated for DS 7.6: times of arrival are calculated in SDP
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: the CTI and time−dependent gain corrections may be applied to CC−mode data (see the Continuous Clocking Mode why topic for details); output filenames include ObsID
25 Apr 2006	added TIMEDEL keyword caveat
01 Dec 2006	updated for CIAO 3.4: CIAO version in errors and warnings
18 Jun 2007	expanded information in Check Source Coordinates section; included software version in which TIMEDEL keyword problem was corrected

URL: http://cxc.harvard.edu/ciao/threads/aciscctoa/ Last modified: 18 Jun 2007