



MIT Kavli Institute



Chandra X-Ray Center

MEMORANDUM

April 23, 2015

To: Jonathan McDowell, SDS Group Leader
From: Glenn E. Allen, SDS
Subject: `acis_process_events` spec
Revision: 3.20
URL: <http://space.mit.edu/CXC/docs/docs.html#ape>
File: `/nfs/inconceivable/d0/SDS/SPECS/acis_process_events/ape_spec_3.20.tex`

1 `acis_process_events`

This spec, which is incomplete, describes some of processing steps for continuous-clocking mode data.

1.1 Description

1.2 Input

1.3 Output

1.4 Parameters

1.5 Processing

1.5.1 Error checking

The following steps are performed once prior to the processing of the data for each input ACIS event.

1. `obsfile`:

(a) Validation:

i. Existence:

If

`obsfile` \neq none and (1)

`obsfile` \neq NONE (2)

and the `obsfile` does not exist, then `obsfile` is changed to “none” and `acis_process_events` produces a warning message.

ii. Permission:

If

$$\text{obsfile} \neq \text{none} \text{ and} \tag{3}$$

$$\text{obsfile} \neq \text{NONE} \tag{4}$$

and the file permissions do not allow the `obsfile` to be read, then `obsfile` is changed to “none” and `acis_process_events` produces a warning message.

iii. OBS_MODE:

If

$$\text{obsfile} \neq \text{none} \text{ and} \tag{5}$$

$$\text{obsfile} \neq \text{NONE}, \tag{6}$$

then

A. If the `obsfile` does not include the keyword `obs_mode`, then `OBS_MODE` is set to “none”.

B. If the `obsfile` includes the keyword `obs_mode` and

$$\text{obs_mode} = \text{pointing} \text{ or} \tag{7}$$

$$\text{obs_mode} = \text{POINTING} \text{ or} \tag{8}$$

$$\text{obs_mode} = \text{secondary} \text{ or} \tag{9}$$

$$\text{obs_mode} = \text{SECONDARY}, \tag{10}$$

then `OBS_MODE` is set to the value of `obs_mode`. Hereafter this keyword is referred to as `OBS_MODE`.

C. If the `obsfile` includes the keyword `obs_mode` and

$$\text{obs_mode} \neq \text{pointing} \text{ and} \tag{11}$$

$$\text{obs_mode} \neq \text{POINTING} \text{ and} \tag{12}$$

$$\text{obs_mode} \neq \text{secondary} \text{ and} \tag{13}$$

$$\text{obs_mode} \neq \text{SECONDARY}, \tag{14}$$

then `OBS_MODE` is set to “none”.

2. infile:

(a) Existence:

If the `infile` does not exist, then `acis_process_events` exits with an error message.

(b) Permission:

If the `infile` exists, but the file permissions do not allow it to be read, then `acis_process_events` exits with an error message.

(c) Validation:

i. OBS_MODE:

If `OBS_MODE = none`, then

A. The `OBS_MODE` is read from the HDU `hin` keyword of the same name. Hereafter this keyword is referred to as `OBS_MODE`.

B. If the HDU `hin` does not include the keyword `OBS_MODE`, then `OBS_MODE` is set to “none” and `acis_process_events` produces a warning message.

C. If the HDU h_{in} includes the keyword `OBS_MODE` and

$$\text{obs_mode} \neq \text{pointing and} \quad (15)$$

$$\text{obs_mode} \neq \text{POINTING and} \quad (16)$$

$$\text{obs_mode} \neq \text{secondary and} \quad (17)$$

$$\text{obs_mode} \neq \text{SECONDARY,} \quad (18)$$

then `OBS_MODE` is set to “none” and `acis_process_events` produces a warning message.

ii. `DATAMODE`:

The `DATAMODE` is read from the HDU h_{in} keyword of the same name. If the HDU h_{in} does not include the keyword `DATAMODE` or if

$$\text{DATAMODE} \neq \text{CC33_FAINT and} \quad (19)$$

$$\text{DATAMODE} \neq \text{CC33_GRADED and} \quad (20)$$

$$\text{DATAMODE} \neq \text{FAINT and} \quad (21)$$

$$\text{DATAMODE} \neq \text{FAINT_BIAS and} \quad (22)$$

$$\text{DATAMODE} \neq \text{GRADED and} \quad (23)$$

$$\text{DATAMODE} \neq \text{VFAINT,} \quad (24)$$

then `acis_process_events` exits with an error message. Hereafter, the value of this keyword is referred to as `DATAMODEin`.

iii. `CONTENT`:

If the `infile` does not have an HDU h_{in} with the keyword

$$\text{CONTENT} = \text{EVT0 or} \quad (25)$$

$$\text{CONTENT} = \text{EVT1 or} \quad (26)$$

$$\text{CONTENT} = \text{TGEVT1 or} \quad (27)$$

$$\text{CONTENT} = \text{EVT2,} \quad (28)$$

then `acis_process_events` exits with an error message. Hereafter, the value of this keyword is referred to as `CONTENTin`.

iv. `TIME`:

If HDU h_{in} of the `infile` does not include the column `TIME`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `TIMEin`.

v. `TIME_R0`:

If

$$\text{DATAMODE}_{in} = \text{CC33_FAINT or} \quad (29)$$

$$\text{DATAMODE}_{in} = \text{CC33_GRADED,} \quad (30)$$

if

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (31)$$

$$\text{CONTENT}_{in} = \text{TGEVT1 or} \quad (32)$$

$$\text{CONTENT}_{in} = \text{EVT2,} \quad (33)$$

and if HDU h_{in} of the `infile` does not include the column `TIME_R0`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `TIME_R0in`.

vi. `EXPNO`:

If HDU h_{in} the `infile` does not include the column `EXPNO`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `EXPNOin`.

vii. CCD_ID:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (34)$$

and if HDU h_{in} of the `infile` does not include the keyword `CCD_ID`, then `acis_process_events` exits with an error message. Hereafter, this keyword is referred to as `CCD_IDin`.

B. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (35)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (36)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (37)$$

and if HDU h_{in} of the `infile` does not include the column `CCD_ID`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `CCD_IDin`.

viii. CCDX:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (38)$$

and if HDU h_{in} of the `infile` does not include the column `CCDX`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `CCDXin`.

ix. CHIPX:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (39)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (40)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (41)$$

and if HDU h_{in} of the `infile` does not include the column `CHIPX`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `CHIPXin`.

x. CCDY:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (42)$$

if

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (43)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (44)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (45)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (46)$$

and if HDU h_{in} of the `infile` does not include the column `CCDY`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `CCDYin`.

xi. TROW:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (47)$$

if

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (48)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (49)$$

and if HDU h_{in} of the `infile` does not include the column `TROW`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `TROWin`.

xii. CHIPY:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (50)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (51)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (52)$$

and if HDU h_{in} of the `infile` does not include the column `CHIPY`, then `acis_process_events` exits with an error message. Hereafter, this column is referred to as `CHIPYin`.

xiii. TIMEDEL:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (53)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (54)$$

and HDU h_{in} of the `infile` does not include the keyword `TIMEDEL`, then `acis_process_events` exits with an error message. Hereafter this keyword is referred to as `TIMEDELin`.

xiv. `RA_TARG`, `DEC_TARG`, `RA_NOM`, `DEC_NOM`, `RA_PNT`, `DEC_PNT`, `CHIPY_TG`, `CHIPY_Z0`, and `TG_M`:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (55)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (56)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (57)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED,} \quad (58)$$

then

A. `RA_TARG`, `DEC_TARG`, `RA_NOM`, `DEC_NOM`, `RA_PNT`, `DEC_PNT`:

If HDU h_{in} of the `infile` does not include the keywords `RA_TARG`, `DEC_TARG`, `RA_NOM`, `DEC_NOM`, `RA_PNT`, and `DEC_PNT`, then `acis_process_events` exits with an error message. Hereafter these keywords are referred to as `RA_TARGin`, `DEC_TARGin`, `RA_NOMin`, `DEC_NOMin`, `RA_PNTin`, and `DEC_PNTin`, respectively.

B. `CHIPY_TG`, `CHIPY_Z0`, and `TG_M`:

If

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (59)$$

and if HDU h_{in} of the `infile` does not include the columns `CHIPY_TG`, `CHIPY_Z0`, and `TG_M`, then `acis_process_events` exits with an error message. Hereafter these columns are referred to as `CHIPY_TGin`, `CHIPY_Z0in`, and `TG_Min`, respectively.

3. `stop`:

(a) Lowercase:

The parameter string is converted to contain only lower case letters.

(b) Validation:

i. Setting:

If

$$\text{stop} \neq \text{none and} \quad (60)$$

stop \neq chip and (61)

stop \neq tdet and (62)

stop \neq det and (63)

stop \neq tan and (64)

stop \neq sky, (65)

then stop is changed to “none” and `acis_process_events` produces a warning message.

ii. OBS_MODE:

If

OBS_MODE \neq pointing and (66)

OBS_MODE \neq POINTING (67)

and

stop \neq none and (68)

stop \neq chip and (69)

stop \neq tdet, (70)

then stop is changed to “none” and `acis_process_events` produces a warning message.

4. `acaofffile`:

(a) Validation for CC mode:

If

OBS_MODE = pointing or (71)

OBS_MODE = POINTING (72)

and

DATAMODE_{in} = CC33_FAINT or (73)

DATAMODE_{in} = CC33_GRADED, (74)

then

i. Setting:

If

acaofffile = none or (75)

acaofffile = NONE, (76)

then `acis_process_events` exits with an error message.

ii. Existence:

If the `acaofffile` does not exist, then `acis_process_events` exits with an error message.

iii. Permission:

If the `acaofffile` exists, but the file permissions do not allow it to be read, then `acis_process_events` exits with an error message.

iv. CONTENT:

If the `acaofffile` does not have an HDU `hacaoff` with the keyword

CONTENT = ASPSOL, (77)

then `acis_process_events` exits with an error message.

- v. Keyword:
If HDU h_{acaoff} of the `acaofffile` does not include the keyword `TSTART`, then `acis_process_events` exits with an error message.
- vi. Columns:
If HDU h_{acaoff} of the `acaofffile` does not include the columns `TIME`, `RA`, `DEC`, and `ROLL` then `acis_process_events` exits with an error message. Hereafter, these columns are referred to as $\text{TIME}_{\text{acaoff}}$, $\text{RA}_{\text{acaoff}}$, $\text{DEC}_{\text{acaoff}}$, and $\text{ROLL}_{\text{acaoff}}$.
- vii. Sequential:
If more than one valid `acaofffile` is specified, but the the values `TSTART` are not in increasing order, then `acis_process_events` exits with an error message.

5. `doevtgrade`:

- (a) Lowercase:
The parameter string is converted to contain only lower case letters.
- (b) Validation:
If

$$\text{doevtgrade} \neq \text{yes and} \tag{78}$$

$$\text{doevtgrade} \neq \text{no}, \tag{79}$$

then `acis_process_events` exits with an error message.

6. `apply_cti`:

- (a) Lowercase:
The parameter string is converted to contain only lower case letters.
- (b) Validation:
 - i. Setting:
If

$$\text{apply_cti} \neq \text{yes and} \tag{80}$$

$$\text{apply_cti} \neq \text{no}, \tag{81}$$

then `acis_process_events` exits with an error message.

- ii. `PHAS`:
If

$$\text{apply_cti} = \text{yes} \tag{82}$$

and if the `infile` does not include the column `PHAS`, then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

- iii. `doevtgrade`:
If

$$\text{apply_cti} = \text{yes and} \tag{83}$$

$$\text{doevtgrade} = \text{no}, \tag{84}$$

then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

7. `alignmentfile`:

(a) Validation for CC mode:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (85)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (86)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (87)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (88)$$

then

i. Setting:

If

$$\text{alignmentfile} = \text{none or} \quad (89)$$

$$\text{alignmentfile} = \text{NONE}, \quad (90)$$

then `acis_process_events` exits with an error message.

ii. Existence:

If the `alignmentfile` does not exist, then `acis_process_events` exits with an error message.

iii. Permission:

If the `alignmentfile` exists, but the file permissions do not allow it to be read, then `acis-process_events` exits with an error message.

iv. CONTENT:

If the `alignmentfile` does not have an HDU $h_{\text{alignment}}$ with the keyword

$$\text{CONTENT} = \text{ASPSOL}, \quad (91)$$

then `acis_process_events` exits with an error message.

v. Keyword:

If HDU $h_{\text{alignment}}$ of the `alignmentfile` does not include the keyword `TSTART`, then `acis-process_events` exits with an error message.

vi. Columns:

If HDU $h_{\text{alignment}}$ of the `alignmentfile` does not include the columns `DY`, `DZ`, and `DTHETA` then `acis_process_events` exits with an error message.

vii. Sequential:

If more than one valid `alignmentfile` is specified, but the the values `TSTART` are not in increasing order, then `acis_process_events` exits with an error message.

8. `badpixfile`:

(a) Validation:

i. Existence:

If

$$\text{badpixfile} \neq \text{none and} \quad (92)$$

$$\text{badpixfile} \neq \text{NONE} \quad (93)$$

and the `badpixfile` does not exist, then `badpixfile` is changed to “none” and `acis-process_events` produces a warning message.

ii. Permission:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (94)$$

$$\text{badpixfile} \neq \text{NONE} \quad (95)$$

and the file permissions do not allow it to be read, then `badpixfile` is changed to “none” and `acis_process_events` produces a warning message.

iii. CONTENT:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (96)$$

$$\text{badpixfile} \neq \text{NONE} \text{ and} \quad (97)$$

and the `badpixfile` does not have one or more HDUs h_{badpix} with the keyword

$$\text{CONTENT} = \text{BADPIX} \text{ or} \quad (98)$$

$$\text{CONTENT} = \text{CDB_ACIS_BADPIX}, \quad (99)$$

then `badpixfile` is changed to “none” and `acis_process_events` produces a warning message.

iv. Keyword:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (100)$$

$$\text{badpixfile} \neq \text{NONE} \quad (101)$$

and the HDU(s) h_{badpix} of the `badpixfile` do not include the keyword `CCD_ID`, then `badpixfile` is changed to “none” and `acis_process_events` produces a warning message. Hereafter this keyword is referred to as `CCD_IDbadpix`.

v. Columns:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (102)$$

$$\text{badpixfile} \neq \text{NONE} \quad (103)$$

and the HDU(s) h_{badpix} of the `badpixfile` do not include the columns `CHIPX`, `CHIPY`, `TIME`, `TIME_STOP`, and `STATUS`, then `badpixfile` is changed to “none” and `acis_process_events` produces a warning message. Hereafter these columns are referred to as `CHIPXbadpix`, `CHIPYbadpix`, `TIMEbadpix`, `TIME_STOPbadpix`, and `STATUSbadpix`, respectively.

9. `ctifile`:

(a) Validation:

If

$$\text{ctifile} \neq \text{caldb} \text{ and} \quad (104)$$

$$\text{ctifile} \neq \text{CALDB}, \quad (105)$$

then

i. Existence:

If the `ctifile` does not exist, then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

ii. Permission:

If the `ctifile` exists, but the file permissions do not allow it to be read, then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

iii. CONTENT:

If the `ctifile` does not have one or more HDUs h_{cti} with the keyword

$$\text{CONTENT} = \text{CDB_ACIS_CTI}, \quad (106)$$

then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

iv. Columns:

If the first such HDU of the `ctifile` does not include the columns `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, `CHIPY_HI`, `PHA`, `VOLUME_X`, `VOLUME_Y`, `FRCTRLX`, `FRCTRLY`, `TCTIX`, and `TCTIY`, then `apply_cti` is changed to “no” and `acis_process_events` produces a warning message.

10. `clobber`:

(a) Lowercase:

The parameter string is converted to contain only lower case letters.

(b) Validation:

i. Setting:

If

$$\text{clobber} \neq \text{yes} \text{ and} \quad (107)$$

$$\text{clobber} \neq \text{no}, \quad (108)$$

then `clobber` is changed to “no” and `acis_process_events` produces a warning message.

ii. Permission:

If

$$\text{clobber} = \text{yes} \quad (109)$$

and the `outfile` exists, but the file permissions of the `outfile` do not allow it to be overwritten, then `acis_process_events` exits with an error message.

iii. Don't overwrite:

If

$$\text{clobber} = \text{no} \quad (110)$$

and the `outfile` exists, then `acis_process_events` exits with an error message.

11. `pix_adj`:

(a) Lowercase:

The parameter string is converted to contain only lower case letters.

(b) Validation:

i. Setting:

If

$$\text{pix_adj} \neq \text{centroid} \text{ and} \quad (111)$$

$$\text{pix_adj} \neq \text{edser} \text{ and} \quad (112)$$

$$\text{pix_adj} \neq \text{none} \text{ and} \quad (113)$$

$$\text{pix_adj} \neq \text{randomize}, \quad (114)$$

then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

ii. OBS_MODE:

If

$$\text{OBS_MODE} \neq \text{pointing and} \quad (115)$$

$$\text{OBS_MODE} \neq \text{POINTING} \quad (116)$$

and

$$\text{pix_adj} \neq \text{none}, \quad (117)$$

then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

iii. stop:

If

$$\text{pix_adj} = \text{centroid or} \quad (118)$$

$$\text{pix_adj} = \text{edser or} \quad (119)$$

$$\text{pix_adj} = \text{randomize} \quad (120)$$

and if

$$\text{stop} \neq \text{sky}, \quad (121)$$

then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

iv. PHAS:

If

$$\text{pix_adj} = \text{centroid} \quad (122)$$

and if the `infile` does not include the column PHAS, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

v. FLTGRADE:

If

$$\text{pix_adj} = \text{edser} \quad (123)$$

and if the `infile` does not include the column FLTGRADE, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

12. `subpixfile`:

(a) If

$$\text{pix_adj} = \text{edser}, \quad (124)$$

then

i. Existence:

If the `subpixfile` does not exist, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

ii. Permission:

If the `subpixfile` exists, but the file permissions do not allow it to be read, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

iii. Validation:

A. CONTENT:

If the `subpixfile` does not have one or more HDUs h_{subpix} with the keyword

$$\text{CONTENT} = \text{AXAF_SUBPIX}, \quad (125)$$

then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

B. Keyword:

If the HDUs h_{subpix} of the `subpixfile` do not include the keyword `CCD_ID`, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message.

C. Columns:

If the HDUs h_{subpix} of the `subpixfile` do not include binary tables with the columns `FLTGRADE`, `NPOINTS`, `ENERGY`, `CHIPX_OFFSET`, and `CHIPY_OFFSET`, then `pix_adj` is changed to “none” and `acis_process_events` produces a warning message. Hereafter these columns are referred to as `FLTGRADEsubpix`, `NPOINTSsubpix`, `ENERGYsubpix`, `CHIPX_OFFSETsubpix`, and `CHIPY_OFFSETsubpix`, respectively.

1.5.2 Initializations

1. Focal-point CCD:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (126)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (127)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (128)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (129)$$

then the values of `RA_PNTin` and `DEC_PNTin` are used to determine the `CCD_ID` associated with the focal point. Hereafter this value is referred to as `CCD_IDfocus*`.

2. Zeroth-order coordinates:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (130)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (131)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (132)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (133)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (134)$$

then the `CHIPY_ZOin` coordinates are processed to obtain the minimum, median, and maximum values:

$$\text{CHIPY_ZO}_{\text{min}} = \text{minimum}(\text{CHIPY_ZO}_{\text{in}}), \quad (135)$$

$$\text{CHIPY_ZO}_{\text{med}} = \text{median}(\text{CHIPY_ZO}_{\text{in}}), \text{ and} \quad (136)$$

$$\text{CHIPY_ZO}_{\text{max}} = \text{maximum}(\text{CHIPY_ZO}_{\text{in}}). \quad (137)$$

*The focal point is the location associated with the optical axis in the absence of dither. This location should not be confused with the aim point, which is the location illuminated by an undithered point source provided that the source is not offset from the target location.

Only events in the good-time intervals are included in the computation of the values of CHIPY_Z0_{\min} , $\text{CHIPY_Z0}_{\text{med}}$, and CHIPY_Z0_{\max} .

3. `acaofffile`:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (138)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (139)$$

and

$$\text{DATAMODE} = \text{CC33_FAINT or} \quad (140)$$

$$\text{DATAMODE} = \text{CC33_GRADED}, \quad (141)$$

then

(a) TIME_{\min} , TIME_{\max} , RA_c , and DEC_c :

The `acaofffile` data are processed to determine the earliest and latest times for which there is aspect information and to determine the right ascension and declination coordinates near the center of the dither pattern:

$$\text{TIME}_{\min} = \min(\text{TIME}_{\text{acaoff}}), \quad (142)$$

$$\text{TIME}_{\max} = \max(\text{TIME}_{\text{acaoff}}), \quad (143)$$

$$\text{RA}_c = \text{median}(\text{RA}_{\text{acaoff}}), \text{ and} \quad (144)$$

$$\text{DEC}_c = \text{median}(\text{DEC}_{\text{acaoff}}). \quad (145)$$

(b) TIME_c :

The `acaofffile` data are processed to determine the time TIME_c at which the quantity

$$\cos(\text{DEC}_{\text{acaoff}}) \cos(\text{DEC}_c) \cos(\text{RA}_{\text{acaoff}} - \text{RA}_c) + \sin(\text{DEC}_{\text{acaoff}}) \sin(\text{DEC}_c) \quad (146)$$

is maximized (i.e. the time at which the telescope is pointed the closest to $(\text{RA}_c, \text{DEC}_c)$).

(c) RA_ADJ_I , DEC_ADJ_I , RA_ADJ_S , DEC_ADJ_S :

The effective values of RA and DEC are computed for the ACIS-I and ACIS-S arrays. These coordinates are used to determine the values of TIME and CHIPY_ADJ .

i. ACIS-I aim point:

For the ACIS-I array, the values of RA_ADJ_I and DEC_ADJ_I are initialized assuming that the source is at the ACIS-I aim point [i.e. that $(\text{TIME}, \text{CCD_ID}, \text{CHIPX}, \text{CHIPY}) = (\text{TIME}_c, 3, 965, 963)^\dagger$].

ii. ACIS-S aim point:

For the ACIS-S array, the values of RA_ADJ_S and DEC_ADJ_S are initialized assuming that the source is at the ACIS-S aim point [i.e. that $(\text{TIME}, \text{CCD_ID}, \text{CHIPX}, \text{CHIPY}) = (\text{TIME}_c, 7, 227, 509)^\ddagger$].

iii. Target location:

For the CCD at the focal point (i.e. $\text{CCD_ID}_{\text{focus}}$), the values of CHIPY are computed for each row of the `acaofffile`, assuming that the source is at the location specified by $\text{RA_TARG}_{\text{in}}$ and $\text{DEC_TARG}_{\text{in}}$. These values of CHIPY are referred to as CHIPY_TARG . If

$$\text{median}(\text{CHIPY_TARG}) \geq 16.5 \text{ and} \quad (147)$$

$$\text{median}(\text{CHIPY_TARG}) < 1008.5, \quad (148)$$

then

[†]As described in the Proposers' Observatory Guide, the location of the aim point on the ACIS-I array has drifted with time. The location used here is within a few dozen pixels of the actual aim point, provided the default `SIM_Y` and `SIM_Z` offsets are used.

[‡]Again, the location used here is within a few dozen pixels of the actual aim point, provided the default `SIM_Y` and `SIM_Z` offsets are used.

A. ACIS-I:

If

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (149)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3, \quad (150)$$

then

$$\text{RA_ADJ}_I = \text{RA_TARG}_{\text{in}} \text{ and} \quad (151)$$

$$\text{DEC_ADJ}_I = \text{DEC_TARG}_{\text{in}}. \quad (152)$$

B. ACIS-S:

If

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (153)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9, \quad (154)$$

then

$$\text{RA_ADJ}_S = \text{RA_TARG}_{\text{in}} \text{ and} \quad (155)$$

$$\text{DEC_ADJ}_S = \text{DEC_TARG}_{\text{in}}. \quad (156)$$

1.5.3 Loop over events

The following steps are performed, in sequence, for each event.

1. STATUS:

(a) Exists:

If HDU h_{in} of the `infile` includes a 32-bit column named `STATUS`, then

- i. The values of the bits for an event are read from the `infile`.
- ii. The value of `STATUS[k]` is set to zero for bits $k = 1-5, 14, 16-19, \text{ and } 23$ (of 0-31), bits that can be set by `acis_process_events`.
- iii. If

$$\text{doevtgrade} = \text{yes}, \quad (157)$$

then the value of `STATUS[20]`, the other bit that can be set by `acis_process_events`, is set to zero.

(b) Does not exist:

If HDU h_{in} does not include a 32-bit column named `STATUS`, then

- i. A set of 32 bits are allocated for the event.
- ii. The values of the 32 bits are initialized to zero.

2. EXPNO:

(a) Read:

The value of `EXPNO` for an event is given by `EXPNOin`.

(b) Validation:

If

$$\text{EXPNO} < 0 \text{ or} \quad (158)$$

$$\text{EXPNO} \geq 10^8, \quad (159)$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. These values should not occur.

3. CCD_ID:

- (a) Read:
The value of CCD_ID for an event is given by $\text{CCD_ID}_{\text{in}}$.
- (b) Validation:
If

$$\text{CCD_ID} < 0 \text{ or} \tag{160}$$

$$\text{CCD_ID} > 9, \tag{161}$$

then `acis_process_events` exits with an error message because CCD_ID-dependent computations could fail if the value of CCD_ID is unphysical.

4. CHIPX:

- (a) Read:
 - i. Level 0:
If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \tag{162}$$

then the value of CHIPX for an event is given by

$$\text{CHIPX} = \text{CCDX}_{\text{in}} + 1. \tag{163}$$

- ii. Level 1, 1.5, or 2:
If

$$\text{CONTENT}_{\text{in}} = \text{EVT1}, \text{ or} \tag{164}$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \text{ or} \tag{165}$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2}, \tag{166}$$

then the value of CHIPX for an event is given by CHIPX_{in} .

- (b) Validation:
 - i. Unphysical:
If

$$\text{CHIPX} < 1 \text{ or} \tag{167}$$

$$\text{CHIPX} > 1024, \tag{168}$$

then `acis_process_events` exits with an error message because CHIPX-dependent computations could fail if the value of CHIPX is unphysical.

- ii. Unexpected:
If

$$\text{CHIPX} = 1 \text{ or} \tag{169}$$

$$\text{CHIPX} = 1024, \tag{170}$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.

5. NODE_ID:

(a) Calculate:

The NODE_ID of an event is given by

$$\text{NODE_ID} = \text{int} \left(\frac{\text{CHIPX} - 1}{256} \right), \quad (171)$$

where “int” means the integer portion of (i.e. truncate or round down) the quantity in parentheses.

6. CHIPY:

(a) Read:

i. Level 0:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (172)$$

then

A. TE mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (173)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (174)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (175)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (176)$$

then the value of CHIPY for an event is given by

$$\text{CHIPY} = \text{CCDY}_{\text{in}} + 1. \quad (177)$$

B. CC mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (178)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (179)$$

then the value of CHIPY for an event is given by

$$\text{CHIPY} = \text{TROW}_{\text{in}} + 1. \quad (180)$$

ii. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (181)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (182)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2}, \quad (183)$$

then the value of CHIPY for an event is given by CHIPY_{in} .

(b) Validation:

i. Unphysical:

A. TE mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (184)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (185)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (186)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \quad (187)$$

and if

$$\text{CHIPY} < 1 \text{ or} \tag{188}$$

$$\text{CHIPY} > 1024, \tag{189}$$

then `acis_process_events` exits with an error message because `CHIPY`-dependent computations could fail if the value of `CHIPY` is unphysical.

B. CC mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \tag{190}$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \tag{191}$$

and if

$$\text{CHIPY} < 1 \text{ or} \tag{192}$$

$$\text{CHIPY} > 512, \tag{193}$$

then `acis_process_events` exits with an error message because the `CHIPY` value is out of range and `CHIPY`-dependent computations could fail if the value of `CHIPY` is unphysical (especially if it is less than 1).

ii. Unexpected:

A. FAINT, FAINT_BIAS, or GRADED:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT} \text{ or} \tag{194}$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS} \text{ or} \tag{195}$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED} \tag{196}$$

and if

$$\text{CHIPY} = 1 \text{ or} \tag{197}$$

$$\text{CHIPY} = 1024, \tag{198}$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.

B. VFAINT:

If

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \tag{199}$$

and if

$$\text{CHIPY} = 1 \text{ or} \tag{200}$$

$$\text{CHIPY} = 2 \text{ or} \tag{201}$$

$$\text{CHIPY} = 1023 \text{ or} \tag{202}$$

$$\text{CHIPY} = 1024, \tag{203}$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or another of these conditions is true. Although these values are not unphysical, they should not occur.

C. CC33_FAINT or CC33_GRADED:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (204)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (205)$$

and if

$$\text{CHIPY} = 1 \text{ or} \quad (206)$$

$$\text{CHIPY} = 512, \quad (207)$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. Although these values are not unphysical, they should not occur.

7. TG_M:

(a) CC mode with gratings:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (208)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (209)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (210)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (211)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (212)$$

then

i. Read:

The value of `TG_M` for an event is given by `TG_Min`.

ii. Validation:

A. If

$$\text{TG_M} < -99, \quad (213)$$

then

$$\text{TG_M} = -99 \quad (214)$$

and `acis_process_events` produces a warning upon completion with a count of the total number of events for which this condition is true. These values should not occur.

B. If

$$\text{TG_M} > 99, \quad (215)$$

then

$$\text{TG_M} = 99 \quad (216)$$

and `acis_process_events` produces a warning upon completion with a count of the total number of events for which this condition is true. These values should not occur.

8. CHIPY_TG:

(a) CC mode with gratings:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (217)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (218)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (219)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (220)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (221)$$

then

i. Read:

The value of CHIPY_TG for an event is given by $\text{CHIPY_TG}_{\text{in}}$.

ii. Validation:

A. If

$$\text{TG_M} > -99 \text{ and} \quad (222)$$

$$\text{TG_M} < 99, \quad (223)$$

and if

$$\text{CHIPY_TG} \leq 0 \text{ or} \quad (224)$$

$$\text{CHIPY_TG} \geq 1025, \quad (225)$$

then `acis_process_events` exits with an error message because CHIPY_TG-dependent computations could fail if the value of CHIPY_TG is unphysical.

B. If

$$\text{TG_M} > -99, \quad (226)$$

$$\text{TG_M} < 99, \text{ and} \quad (227)$$

$$\text{CHIPY_TG} < 1, \quad (228)$$

then

$$\text{CHIPY_TG} = 1. \quad (229)$$

C. If

$$\text{TG_M} > -99, \quad (230)$$

$$\text{TG_M} < 99, \text{ and} \quad (231)$$

$$\text{CHIPY_TG} > 1024, \quad (232)$$

then

$$\text{CHIPY_TG} = 1024. \quad (233)$$

9. CHIPY_Z0:

(a) CC mode with gratings:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (234)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (235)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (236)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (237)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (238)$$

then

i. Read:

The value of CHIPY_Z0 for an event is given by $\text{CHIPY_Z0}_{\text{in}}$.

10. TIME_RO:

(a) CC mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (239)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (240)$$

then

i. Read:

A. Level 0:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (241)$$

then the value of TIME_RO for an event is given by TIME_{in} .

B. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1, or} \quad (242)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1, or} \quad (243)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2}, \quad (244)$$

then the value of TIME_RO for an event is given by $\text{TIME_RO}_{\text{in}}$.

ii. Validation:

If

$$\text{TIME_RO} < 0 \text{ or} \quad (245)$$

$$\text{TIME_RO} \geq 3 \times 10^9, \quad (246)$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. These values should not occur.

11. TIME and CHIPY_ADJ:

(a) Read or calculate:

i. TE mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (247)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (248)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (249)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT,} \quad (250)$$

then

$$\text{TIME} = \text{TIME}_{\text{in}} \text{ and} \quad (251)$$

$$\text{CHIPY_ADJ} = \text{CHIPY.} \quad (252)$$

ii. Pointing CC mode without grating data:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (253)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (254)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (255)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (256)$$

and

$$\text{CONTENT}_{\text{in}} = \text{EVT0 or} \quad (257)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (258)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2,} \quad (259)$$

then

A. TIME':

The approximate time of arrival

$$\text{TIME}' = \text{TIME}_{\text{RO}} - (512 + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (260)$$

B. CHIPY_ADJ':

If

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (261)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3, \quad (262)$$

then CHIPY_ADJ' (the approximate value of CHIPY_ADJ) is given by the CHIPY location (on the focal-point CCD) of the coordinates RA_ADJ_I and DEC_ADJ_I at the time TIME'. If TIME' < TIME_{min} or TIME' > TIME_{max}, then TIME_c is used instead of TIME'.

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (263)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9, \quad (264)$$

then CHIPY_ADJ' is given by the CHIPY location (on the focal-point CCD) of the coordinates RA_ADJ_S and DEC_ADJ_S at the time TIME'. If TIME' < TIME_{min} or TIME' > TIME_{max}, then TIME_c is used instead of TIME'.

C. TIME:

The value of CHIPY_ADJ' is used to obtain a better estimate of the time of arrival

$$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ}' + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (265)$$

D. CHIPY_ADJ:

If

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (266)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3, \quad (267)$$

then the value of CHIPY_ADJ is given by the CHIPY location (on the focal-point CCD) of the coordinates RA_ADJ_I and DEC_ADJ_I at the time TIME. If TIME < TIME_{min} or TIME > TIME_{max}, then TIME_c is used instead of TIME. If

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (268)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9, \quad (269)$$

then the value of CHIPY_ADJ is given by the CHIPY location (on the focal-point CCD) of the coordinates RA_ADJ_S and DEC_ADJ_S at the time TIME. If TIME < TIME_{min} or TIME > TIME_{max}, then TIME_c is used instead of TIME.

iii. Pointing CC mode with ACIS-S grating data:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (270)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (271)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (272)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (273)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (274)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (275)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9, \quad (276)$$

then

A. Source events:

If

$$\text{TG_M} > -99 \text{ and} \quad (277)$$

$$\text{TG_M} < 99, \quad (278)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_TG and} \quad (279)$$

$$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ} + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (280)$$

If the event does not occur during a good-time interval, then CHIPY_ADJ = CHIPY_ZO_{med} instead of CHIPY_TG in equation 279.

B. Background events with zeroth order on the array:

If

$$\text{TG_M} = -99 \text{ or} \quad (281)$$

$$\text{TG_M} = 99 \quad (282)$$

and if

$$\text{CHIPY_Z0}_{\min} \geq 0.5 \text{ and} \quad (283)$$

$$\text{CHIPY_Z0}_{\max} < 1024.5, \quad (284)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_Z0} \text{ and} \quad (285)$$

$$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ} + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (286)$$

If the event does not occur during a good-time interval, then $\text{CHIPY_ADJ} = \text{CHIPY_Z0}_{\text{med}}$ instead of CHIPY_Z0 in equation. 285.

C. Background events with zeroth order off the array:

If

$$\text{TG_M} = -99 \text{ or} \quad (287)$$

$$\text{TG_M} = 99 \quad (288)$$

and if

$$\text{CHIPY_Z0}_{\max} < 0.5 \text{ or} \quad (289)$$

$$\text{CHIPY_Z0}_{\min} \geq 1024.5, \quad (290)$$

then

$$\text{CHIPY_ADJ} = 512 + (\text{CHIPY_Z0} - \text{CHIPY_Z0}_{\text{med}}) \text{ and} \quad (291)$$

$$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ} + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (292)$$

If the event does not occur during a good-time interval, then $\text{CHIPY_ADJ} = 512$ instead of $512 + \text{CHIPY_Z0} - \text{CHIPY_Z0}_{\text{med}}$ in equation. 291.

iv. Pointing CC mode with ACIS-I grating data:

If

$$\text{OBS_MODE} = \text{pointing} \text{ or} \quad (293)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (294)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (295)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (296)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (297)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (298)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3, \quad (299)$$

then

- A. `TIME'`:
The approximate time of arrival

$$\text{TIME}' = \text{TIME_RO} - (512 + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (300)$$

- B. `CHIPY_ADJ'`:
`CHIPY_ADJ'` (the approximate value of `CHIPY_ADJ`) is given by the `CHIPY` location (on the focal-point CCD) of the coordinates `RA_ADJI` and `DEC_ADJI` at the time `TIME'`. If the event does not occur during a good-time interval, then `TIMEc` is used instead of `TIME'`.

- C. `TIME`:
The value of `CHIPY_ADJ'` is used to obtain a better estimate of the time of arrival

$$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ}' + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (301)$$

- D. `CHIPY_ADJ`:
The value of `CHIPY_ADJ` is given by the `CHIPY` location (on the focal-point CCD) of the coordinates `RA_ADJI` and `DEC_ADJI` at the time `TIME`. If the event does not occur during a good-time interval, then `TIMEc` is used instead of `TIME`.

- v. Secondary CC mode:
If

$$\text{OBS_MODE} \neq \text{pointing and} \quad (302)$$

$$\text{OBS_MODE} \neq \text{POINTING}, \quad (303)$$

then

- A. `TIME`:

$$\text{TIME} = \text{TIME_RO} - (512 + 1028) \times \text{TIMEDEL}_{\text{in}}. \quad (304)$$

- B. `CHIPY_ADJ`:

$$\text{CHIPY_ADJ} = 512. \quad (305)$$

- (b) Validation:

- i. If

$$\text{TIME} < 0 \text{ or} \quad (306)$$

$$\text{TIME} \geq 3 \times 10^9, \quad (307)$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which one or the other of these conditions is true. These values should not occur.

- ii. If

$$\text{CHIPY_ADJ} < 0.5 \text{ or} \quad (308)$$

$$\text{CHIPY_ADJ} \geq 1024.5, \quad (309)$$

then `acis_process_events` exits with an error message because `CHIPY_ADJ`-dependent computations could fail if the value of `CHIPY_ADJ` is unphysical.

12. Bad pixel:

(a) If

$$\text{badpixfile} \neq \text{none and} \quad (310)$$

$$\text{badpixfile} \neq \text{NONE} \quad (311)$$

and if the `badpixfile` includes a valid HDU h_{badpix} where $\text{CCD_ID}_{\text{badpix}} = \text{CCD_ID}$, then the HDU h_{badpix} is searched as follows to determine if the event should have one or more `STATUS` bits set to one.

- i. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (312)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (313)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (314)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (315)$$

and where

$$\text{STATUS}_{\text{badpix},r}[5] = 1 \text{ or} \quad (316)$$

$$\text{STATUS}_{\text{badpix},r}[6] = 1 \text{ or} \quad (317)$$

$$\text{STATUS}_{\text{badpix},r}[9] = 1, \quad (318)$$

then

$$\text{STATUS}[0] = 1 \quad (319)$$

for the event. Here $\text{CCD_ID}_{\text{badpix}}$ is the value of the keyword `CCD_ID` in HDU h_{badpix} of the `badpixfile`, $\text{CHIPX}_{\text{badpix},r}[0]$ and $\text{CHIPX}_{\text{badpix},r}[1]$ are the first and second values in the vector column named `CHIPX` of row r of HDU h_{badpix} of the `badpixfile`, and $\text{TIME}_{\text{badpix},r}$ and $\text{TIME_STOP}_{\text{badpix},r}$ are the values in the columns named `TIME` and `TIME_STOP`, respectively, of row r of HDU h_{badpix} of the `badpixfile`.

- ii. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (320)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (321)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (322)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (323)$$

and where

$$\text{STATUS}_{\text{badpix},r}[0] = 1 \text{ or} \quad (324)$$

$$\text{STATUS}_{\text{badpix},r}[1] = 1 \text{ or} \quad (325)$$

$$\text{STATUS}_{\text{badpix},r}[7] = 1 \text{ or} \quad (326)$$

$$\text{STATUS}_{\text{badpix},r}[11] = 1 \text{ or} \quad (327)$$

$$\text{STATUS}_{\text{badpix},r}[12] = 1 \text{ or} \quad (328)$$

$$\text{STATUS}_{\text{badpix},r}[13] = 1 \text{ or} \quad (329)$$

$$\text{STATUS}_{\text{badpix},r}[14] = 1 \text{ or} \quad (330)$$

$$\text{STATUS}_{\text{badpix},r}[16] = 1, \quad (331)$$

then

$$\text{STATUS}[4] = 1 \quad (332)$$

for the event.

- iii. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (333)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (334)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (335)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (336)$$

and where

$$\text{STATUS}_{\text{badpix},r}[8] = 1 \text{ or} \quad (337)$$

$$\text{STATUS}_{\text{badpix},r}[10] = 1, \quad (338)$$

then

$$\text{STATUS}[5] = 1 \quad (339)$$

for the event.

- iv. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (340)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (341)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (342)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (343)$$

and where

$$\text{STATUS}_{\text{badpix},r}[3] = 1, \quad (344)$$

then

$$\text{STATUS}[6] = 1 \quad (345)$$

for the event.

- v. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (346)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (347)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (348)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (349)$$

and where

$$\text{STATUS}_{\text{badpix},r}[2] = 1 \text{ or} \quad (350)$$

$$\text{STATUS}_{\text{badpix},r}[4] = 1, \quad (351)$$

then

$$\text{STATUS}[8] = 1 \quad (352)$$

for the event.

- vi. If $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$ or $\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}$ and if there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0], \quad (353)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1], \quad (354)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r}, \quad (355)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (356)$$

and where

$$\text{STATUS}_{\text{badpix},r}[15] = 1, \quad (357)$$

then

$$\text{STATUS}[16] = 1 \quad (358)$$

for the event.

- vii. In summary, the mapping between a bad-pixel STATUS bit and the corresponding event STATUS bit is listed in Table 1.

Table 1: Bad-pixel to event STATUS bit mapping

Bad-pixel STATUS bit	Event STATUS bit
0	4
1	4
2	8
3	6
4	8
5	0
6	0
7	4
8	5
9	0
10	5
11	4
12	4
13	4
14	4
15	16
16	4

13. PHAS:

- (a) If HDU 1 of the `infile` includes the column `PHAS`, then
- i. the values of `PHAS` for an event are read from the `infile`.
 - ii. If $\text{PHAS}[4] < \text{split threshold}$, then $\text{STATUS}[k] = 1$ for bit $k = 1$.
 - iii. If $\text{PHAS}[4] \leq \text{PHAS}[j]$, for one or more $j = 0-3$ or $5-8$, then $\text{STATUS}[k] = 1$ for bit $k = 1$.
 - iv. If $\text{PHAS}[j] > 4095$, for one or more $j = 0-8$, then $\text{STATUS}[k] = 1$ for bit $k = 2$.

14. PHAS_ADJ:

- (a) If HDU 1 of the `infile` includes $\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}$, if the parameter `apply_cti` = yes, and if the `ctifile` and `mtlfile` are specified, then the CTI-adjusted pulse heights are computed as follows.

- i. The real-valued arrays for the serial CTI adjustment Δ_x , the parallel CTI adjustment Δ_y , and the adjusted pulse heights PHAS_ADJ are initialized such that

$$\Delta_x[j] = 0, \quad (359)$$

$$\Delta_y[j] = 0, \text{ and} \quad (360)$$

$$\text{PHAS_ADJ}[j] = \text{PHAS}[j] \quad (361)$$

for every element $j = 0-8$, where the starting point for the adjusted pulse heights are the unadjusted pulse heights PHAS. Note that the values of the unadjusted pulse heights PHAS remain unchanged to ensure that it is possible to remove the CTI adjustment or to reapply the adjustment if the algorithm or calibration data are modified.

- ii. The CTI iteration counter n is initialized such that

$$n = 1. \quad (362)$$

- iii. The temporary variables Δ'_x , Δ'_y , and PHAS_ADJ' are set such that

$$\Delta'_x[j] = \Delta_x[j], \quad (363)$$

$$\Delta'_y[j] = \Delta_y[j], \text{ and} \quad (364)$$

$$\text{PHAS_ADJ}'[j] = \text{PHAS_ADJ}[j] \quad (365)$$

for each element j .

- iv. A. If there is a serial CTI trap-density map in the `ctifile` for `CCD_ID` and if `NODE_ID = 0` or `2`, then the values of Δ_x are given by

$$\Delta_x[0] = c_x[0]s_x\rho_x[0]V_x[0], \quad (366)$$

$$\Delta_x[1] = c_x[1]s_x\rho_x[1]V_x[1] - c'_x[0]s_x\rho_x[0]V_x[0], \quad (367)$$

$$\Delta_x[2] = c_x[2]s_x\rho_x[2]V_x[2] - c'_x[1]s_x\rho_x[1]V_x[1], \quad (368)$$

$$\Delta_x[3] = c_x[3]s_x\rho_x[3]V_x[3], \quad (369)$$

$$\Delta_x[4] = c_x[4]s_x\rho_x[4]V_x[4] - c'_x[3]s_x\rho_x[3]V_x[3], \quad (370)$$

$$\Delta_x[5] = c_x[5]s_x\rho_x[5]V_x[5] - c'_x[4]s_x\rho_x[4]V_x[4], \quad (371)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6], \quad (372)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[6]s_x\rho_x[6]V_x[6], \text{ and} \quad (373)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8] - c'_x[7]s_x\rho_x[7]V_x[7]. \quad (374)$$

where

$$c_x[j] = \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j\text{)}, \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \\ \text{(for } j = 1, 2, 4, 5, 7, 8\text{)}, \end{array} \right. \\ \text{FRCTRLX} \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 3, 6\text{)} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \\ \text{(for } j = 1, 2, 4, 5, 7, 8\text{)}, \end{array} \right. \end{cases}$$

$$c'_x[j] = \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] < \text{split threshold or} \\ j \rightarrow \text{CHIPX} = 1, 256, 513, \text{ or } 768 \\ \text{(for } j = 0, 1, 3, 4, 6, 7), \end{array} \right. \\ \text{FRCTRLX} & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \text{ and} \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 3, 4, 6, 7), \end{array} \right. \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 3, 4, 6, 7), \end{array} \right. \end{cases}$$

$$s_x = 1 + \text{TCTIX}(T - \text{FP_TEMPO}), \quad (375)$$

$$\begin{cases} s_x \text{ is a temperature dependent scaling factor,} \\ \text{TCTIX is the CCD_ID dependent value in the column TCTIX of the} \\ \text{ctifile,} \\ \text{FP_TEMPO is the name of a keyword in the ctifile,} \end{cases}$$

$$T = \left(\frac{t' - t'_k}{t'_{k+1} - t'_k} \right) (\text{FP_TEMP}_{k+1} - \text{FP_TEMP}_k) + \text{FP_TEMP}_k, \quad (376)$$

$$\begin{cases} T \text{ is the time dependent focal plane temperature,} \\ t' = t + \text{TIMEDEL}_{\text{in}} (\text{TIMEPIXR}_{\text{evt}} - 0.5), \end{cases} \quad (377)$$

$$\begin{cases} t \text{ is the TIME of the event,} \\ \text{TIMEPIXR}_{\text{evt}} \text{ is a keyword in the infile,} \end{cases}$$

$$t'_k = \text{TIME}_k + \text{TIMEDEL}_{\text{mtl}} (\text{TIMEPIXR}_{\text{mtl}} - 0.5), \quad (378)$$

$$\begin{cases} \text{TIME}_k \text{ is the } k^{\text{th}} \text{ element of the column TIME in the mtlfile,} \\ t'_k \leq t', \\ \text{If } t' < t'_k \text{ for } k = 0, \text{ then } k = 0, \\ \text{FP_TEMP}_k \text{ is the } k^{\text{th}} \text{ element of the column FP_TEMP in the mtlfile,} \\ \text{TIMEDEL}_{\text{mtl}} \text{ is a keyword in the mtlfile,} \\ \text{TIMEPIXR}_{\text{mtl}} \text{ is a keyword in the mtlfile,} \end{cases}$$

$$t'_{k+1} = \text{TIME}_{k+1} + \text{TIMEDEL}_{\text{mtl}} (\text{TIMEPIXR}_{\text{mtl}} - 0.5), \quad (379)$$

$$\begin{cases} \text{TIME}_{k+1} \text{ is the } (k+1)^{\text{th}} \text{ element of the column TIME in the mtlfile,} \\ t'_{k+1} > t', \\ \text{If } t' > t'_k \text{ for } k = n, \text{ where } n \text{ is the last element, then } k = n, \\ \text{FP_TEMP}_{k+1} \text{ is the } (k+1)^{\text{th}} \text{ element of the column FP_TEMP in the} \\ \text{mtlfile,} \end{cases}$$

$$\rho_x[j] = \text{serial trap density}, \quad (380)$$

$$\begin{cases} \rho_x[j] \text{ depends upon the CCD_ID and upon the CHIPX and nint(CHIPY_ADJ)} \\ \text{coordinates associated with element } j \text{ of PHAS_ADJ}[j] \text{ (see Fig. 1),} \end{cases}$$

$$V_x[j] = \left(\frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME_X}_{l+1} - \text{VOLUME_X}_l) + \text{VOLUME_X}_l, \quad (381)$$

$$\left\{ \begin{array}{l} \text{PHA}_l \text{ is the } l^{\text{th}} \text{ element of the column PHA in the ctifile,} \\ \text{PHA}_l \text{ (and PHA}_{l+1}) \text{ are CCD_ID dependent,} \\ \text{PHA}_l \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j], \\ \text{If } \text{PHA}_l > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \text{ for } l = 0, \text{ then } l = 0, \\ \text{PHA}_{l+1} \text{ is the } (l+1)^{\text{th}} \text{ element of the column PHA in the ctifile,} \\ \text{PHA}_{l+1} > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j], \\ \text{If } \text{PHA}_{l+1} \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \text{ for } l = n, \text{ where } n \text{ is the last} \\ \text{element, then } l = n, \\ \text{VOLUME_X}_l \text{ is the } l^{\text{th}} \text{ element of the column VOLUME_X in the ctifile,} \\ \text{VOLUME_X}_l, \text{ which is CCD_ID dependent, is associated with } \text{PHA}_l, \\ \text{VOLUME_X}_{l+1} \text{ is the } (l+1)^{\text{th}} \text{ element of the column VOLUME_X in the} \\ \text{ctifile, and} \\ \text{VOLUME_X}_{l+1}, \text{ which is CCD_ID dependent, is associated with } \text{PHA}_{l+1} \end{array} \right.$$

B. If there is a serial CTI trap-density map in the ctifile for CCD_ID and if NODE_ID = 1 or 3, then the values of Δ_x are given by

$$\Delta_x[0] = c_x[0]s_x\rho_x[0]V_x[0] - c'_x[1]s_x\rho_x[1]V_x[1], \quad (382)$$

$$\Delta_x[1] = c_x[1]s_x\rho_x[1]V_x[1] - c'_x[2]s_x\rho_x[2]V_x[2], \quad (383)$$

$$\Delta_x[2] = c_x[2]s_x\rho_x[2]V_x[2], \quad (384)$$

$$\Delta_x[3] = c_x[3]s_x\rho_x[3]V_x[3] - c'_x[4]s_x\rho_x[4]V_x[4], \quad (385)$$

$$\Delta_x[4] = c_x[4]s_x\rho_x[4]V_x[4] - c'_x[5]s_x\rho_x[5]V_x[5], \quad (386)$$

$$\Delta_x[5] = c_x[5]s_x\rho_x[5]V_x[5], \quad (387)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6] - c'_x[7]s_x\rho_x[7]V_x[7], \quad (388)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[8]s_x\rho_x[8]V_x[8], \text{ and} \quad (389)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8]. \quad (390)$$

where

$$c_x[j] = \left\{ \begin{array}{ll} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \\ \text{(for } j = 0, 1, 3, 4, 6, 7), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 2, 5, 8) \end{array} \right. \\ \text{FRCTRLX} & \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \\ \text{(for } j = 0, 1, 3, 4, 6, 7), \end{array} \right. \end{array} \right.$$

$$c'_x[j] = \left\{ \begin{array}{l} 0 \\ \text{FRCTRLX} \\ 1 \end{array} \right. \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] < \text{split threshold or} \\ j \rightarrow \text{CHIPX} = 257, 512, 769, \text{ or } 1024 \\ \text{(for } j = 1, 2, 4, 5, 7, 8), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \text{ and} \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \geq \text{split threshold} \\ \text{(for } j = 1, 2, 4, 5, 7, 8), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 1, 2, 4, 5, 7, 8), \end{array} \right.$$

and s_x , T , t' , t'_k , t'_{k+1} , $\rho_x[j]$, and $V_x[j]$ are given by equations. 375, 376, 377, 378, 379, 380, and 381, respectively.

- v. If there is a parallel CTI trap-density map in the `ctifile` for `CCD_ID`, then the values of Δ_y are given by

$$\Delta_y[0] = c_y[0]s_y\rho_y[0]V_y[0], \quad (391)$$

$$\Delta_y[1] = c_y[1]s_y\rho_y[1]V_y[1], \quad (392)$$

$$\Delta_y[2] = c_y[2]s_y\rho_y[2]V_y[2], \quad (393)$$

$$\Delta_y[3] = c_y[3]s_y\rho_y[3]V_y[3] - c'_y[0]s_y\rho_y[0]V_y[0], \quad (394)$$

$$\Delta_y[4] = c_y[4]s_y\rho_y[4]V_y[4] - c'_y[1]s_y\rho_y[1]V_y[1], \quad (395)$$

$$\Delta_y[5] = c_y[5]s_y\rho_y[5]V_y[5] - c'_y[2]s_y\rho_y[2]V_y[2], \quad (396)$$

$$\Delta_y[6] = c_y[6]s_y\rho_y[6]V_y[6] - c'_y[3]s_y\rho_y[3]V_y[3], \quad (397)$$

$$\Delta_y[7] = c_y[7]s_y\rho_y[7]V_y[7] - c'_y[4]s_y\rho_y[4]V_y[4], \quad (398)$$

$$\Delta_y[8] = c_y[8]s_y\rho_y[8]V_y[8] - c'_y[5]s_y\rho_y[5]V_y[5], \quad (399)$$

where

$$c_y[j] = \left\{ \begin{array}{l} 0 \\ \text{FRCTRLX} \\ 1 \end{array} \right. \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8), \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2) \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8), \end{array} \right.$$

$$c'_y[j] = \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] < \text{split threshold or} \\ j \rightarrow \text{CHIPY} = 1 \text{ or } 1024 \\ \text{(for } j = 1, 2, 3, 4, 5), \end{array} \right. \\ \text{FRCTRLY} & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5), \end{array} \right. \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5), \end{array} \right. \end{cases}$$

$$s_y = 1 + \text{TCTIY}(T - \text{FP_TEMPO}), \quad (400)$$

$$\left\{ \begin{array}{l} s_y \text{ is a temperature dependent scaling factor,} \\ \text{TCTIY is the CCD_ID dependent value in the column TCTIY of the} \\ \text{ctifile,} \\ \text{FP_TEMPO is the name of a keyword in the ctifile,} \end{array} \right.$$

$$\rho_y[j] = \text{parallel trap density}, \quad (401)$$

$$\left\{ \begin{array}{l} \rho_y[j] \text{ depends upon the CCD_ID and upon the CHIPX and nint(CHIPY_ADJ)} \\ \text{coordinates associated with element } j \text{ of PHAS_ADJ}[j] \text{ (see Fig. 1),} \end{array} \right.$$

$$V_y[j] = \left(\frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME_Y}_{l+1} - \text{VOLUME_Y}_l) + \text{VOLUME_Y}_l, \quad (402)$$

$$\left\{ \begin{array}{l} \text{PHA}_l \text{ is the } l^{\text{th}} \text{ element of the column PHA in the ctifile,} \\ \text{PHA}_l \text{ (and } \text{PHA}_{l+1}) \text{ are CCD_ID dependent,} \\ \text{PHA}_l \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j], \\ \text{If } \text{PHA}_l > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \text{ for } l = 0, \text{ then } l = 0, \\ \text{PHA}_{l+1} \text{ is the } (l+1)^{\text{th}} \text{ element of the column PHA in the ctifile,} \\ \text{PHA}_{l+1} > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j], \\ \text{If } \text{PHA}_{l+1} \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \text{ for } l = n, \text{ where } n \text{ is the last} \\ \text{element, then } l = n, \\ \text{VOLUME_Y}_l \text{ is the } l^{\text{th}} \text{ element of the column VOLUME_Y in the ctifile,} \\ \text{VOLUME_Y}_l, \text{ which is CCD_ID dependent, is associated with } \text{PHA}_l, \\ \text{VOLUME_Y}_{l+1} \text{ is the } (l+1)^{\text{th}} \text{ element of the column VOLUME_Y in the} \\ \text{ctifile,} \\ \text{VOLUME_Y}_{l+1}, \text{ which is CCD_ID dependent, is associated with } \text{PHA}_{l+1}, \end{array} \right.$$

and T , t' , t'_k , and t'_{k+1} , are given by equations. 376, 377, 378, and 379, respectively.

vi. The CTI-adjusted pulse heights

$$\text{PHAS_ADJ}[j] = \text{PHAS}[j] + \Delta_x[j] + \Delta_y[j] \quad (403)$$

for all j .

vii. A. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| < \text{cticonverge} \text{ (for all } j) \text{ and} \quad (404)$$

$$n \leq \text{max_cti_iter}, \quad (405)$$

then the computation of PHAS_ADJ is complete for the event.

B. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| \geq \text{cticonverge (for one or more } j) \text{ and} \quad (406)$$

$$n < \text{max_cti_iter}, \quad (407)$$

then $n = n + 1$ and steps 1.5.14(a)iii–1.5.14(a)vii are repeated.

C. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| \geq \text{cticonverge (for one or more } j) \text{ and} \quad (408)$$

$$n \geq \text{max_cti_iter}, \quad (409)$$

then no additional iterations are performed, the values of $\text{PHAS_ADJ}[j]$ from the most recent iteration are used as are, and $\text{STATUS}[k] = 1$ for bit $k = 20$ to indicate that the CTI adjustment did not converge.

15. FLTGRADE:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ and} \quad (410)$$

$$\text{apply_cti} = \text{yes}, \quad (411)$$

then

$$\text{FLTGRADE} = c_f[0] + 2c_f[1] + 4c_f[2] + 8c_f[3] + 16c_f[5] + 32c_f[6] + 64c_f[7] + 128c_f[8], \quad (412)$$

where

$$c_f[j] = \begin{cases} 0 & \text{if } \text{PHAS_ADJ}[j] < \text{split threshold} \\ 1 & \text{otherwise,} \end{cases} \quad (413)$$

and the elements $j = 0-3$ and $5-8$ of PHAS_ADJ are depicted in Figure 1.

(b) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ and} \quad (414)$$

$$\text{apply_cti} = \text{no}, \quad (415)$$

then

$$\text{FLTGRADE} = c_f[0] + 2c_f[1] + 4c_f[2] + 8c_f[3] + 16c_f[5] + 32c_f[6] + 64c_f[7] + 128c_f[8], \quad (416)$$

where

$$c_f[j] = \begin{cases} 0 & \text{if } \text{PHAS}[j] < \text{split threshold} \\ 0 & \text{if } \text{PHAS}[j] > 4095 \\ 0 & \text{if } \text{PHAS}[j] > \text{PHAS}[4] \text{ for } j = 0-3 \\ 0 & \text{if } \text{PHAS}[j] \geq \text{PHAS}[4] \text{ for } j = 5-8 \\ 1 & \text{otherwise.} \end{cases} \quad (417)$$

(c) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (418)$$

then the FLTGRADE of an event is equal to the value of FLTGRADE for the event in the `infile`.

16. GRADE:

(a) If the `gradefile` is specified, then the GRADE of an event is determined from the FLTGRADE of the event as follows.

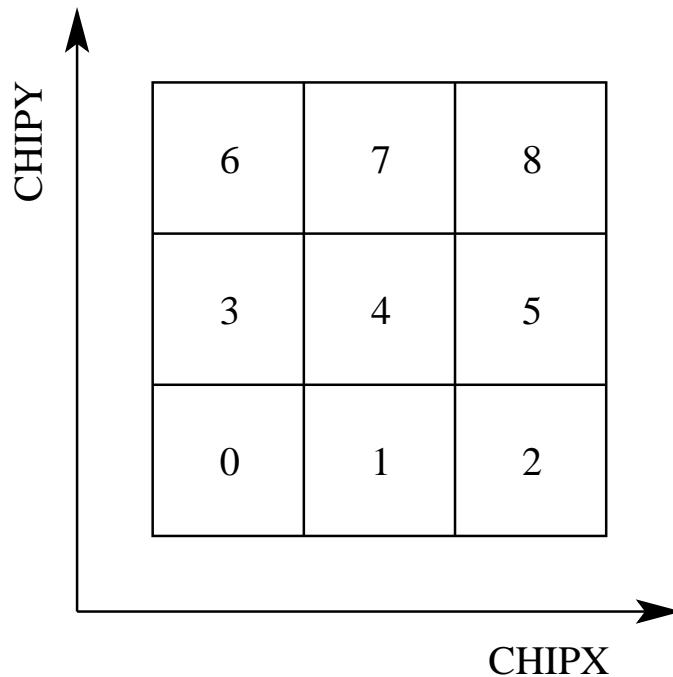


Figure 1: The relative CHIPX and CHIPY coordinates of the nine elements $j = 0-8$ of a $3 \text{ pixel} \times 3 \text{ pixel}$ event island PHAS[j] or PHAS_ADJ[j].

- i. The appropriate HDU of the `grade` file is identified. This HDU is the one where the header keyword `CBD10001` includes the `DATAMODEin` of HDU 1 of the `infile`.
- ii. The row i of the appropriate HDU of the `grade` file is identified. This row is the one where

$$\text{FLTGRADE}_{\text{grade},i} = \text{FLTGRADE}, \quad (419)$$

where `FLTGRADEgrade` is a column in the `grade` file.

- iii. The `GRADE` of the event is given by

$$\text{GRADE} = \text{GRADE}_{\text{grade},i}, \quad (420)$$

where `GRADEgrade` is a column in the `grade` file.

17. PHA_RO:

- (a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}, \quad (421)$$

then

$$\text{PHA_RO} = \sum_{j=0}^8 \beta[j] p[j], \quad (422)$$

where

- i.

$$p[j] = \text{PHAS}[j]. \quad (423)$$

- ii. The elements $j = 0-8$ of PHAS are depicted in Figure 1.

iii.

$$\beta[j] = 0 \quad \text{if} \quad p[j] < \text{split threshold.} \quad (424)$$

iv.

$$\beta[j] = 0 \quad \text{if} \quad \begin{cases} p[j] > p[4] & (\text{for } j = 0-3) \\ p[j] \geq p[4] & (\text{for } j = 5-8) \end{cases} \quad (425)$$

v. If `CORNERS` = -1, then

$$\beta[0] = \beta[2] = \beta[6] = \beta[8] = 0. \quad (426)$$

vi. If `CORNERS` = 0, then there are no additional constraints on $\beta[0]$, $\beta[2]$, $\beta[6]$, and $\beta[8]$.

vii. If `CORNERS` = 1, then

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[3] = 0. \quad (427)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[5] = 0. \quad (428)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (429)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (430)$$

viii. If `CORNERS` = 2, then

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[3] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (431)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[5] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (432)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (433)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (434)$$

(b) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (435)$$

then

i. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (436)$$

then the value of `PHA_RO` for the event is the value of `PHA` in the `infile`.

ii. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1}, \text{ TGEVT1}, \text{ or } \text{EVT2}, \quad (437)$$

then the value of `PHA_RO` for the event is the value of `PHA_RO` in the `infile`.

18. `PHA`, including time-dependent gain:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}, \quad (438)$$

then

$$\text{PHA} = \sum_{j=0}^8 \beta[j] p[j], \quad (439)$$

where

i.

$$p[j] = \begin{cases} \text{PHAS_ADJ}[j] & \text{if } \text{apply_cti} = \text{yes} \\ \text{PHAS}[j] & \text{if } \text{apply_cti} = \text{no} \end{cases} \quad (440)$$

ii. The elements $j = 0-8$ of `PHAS_ADJ` (or `PHAS`) are depicted in Figure 1.

iii.

$$\beta[j] = 0 \quad \text{if} \quad p[j] < \text{split threshold.} \quad (441)$$

iv. If the CTI adjustment is not performed, then

$$\beta[j] = 0 \quad \text{if} \quad \begin{cases} p[j] > p[4] & (\text{for } j = 0-3) \\ p[j] \geq p[4] & (\text{for } j = 5-8) \end{cases} \quad (442)$$

v. If CORNERS = -1, then

$$\beta[0] = \beta[2] = \beta[6] = \beta[8] = 0. \quad (443)$$

vi. If CORNERS = 0, then there are no additional constraints on $\beta[0]$, $\beta[2]$, $\beta[6]$, and $\beta[8]$.

vii. If CORNERS = 1, then

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[3] = 0. \quad (444)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[5] = 0. \quad (445)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (446)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (447)$$

viii. If CORNERS = 2, then

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[3] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (448)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[5] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (449)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (450)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (451)$$

(b) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (452)$$

then the value of PHA for the event is read from the `infile`.

(c) If

$$\text{apply_tgain} = \text{yes}, \quad (453)$$

then

$$\text{PHA} = \text{PHA} + \text{int} \left[\left(\frac{\text{TIME} - \text{EPOCH1}}{\text{EPOCH2} - \text{EPOCH1}} \right) (\delta_2 - \delta_1) + \delta_1 + \epsilon \right], \quad (454)$$

where

$$\text{int} = \text{the integer portion of (i.e. truncate or round down)}, \quad (455)$$

$$\text{TIME} = \text{the time of the event}, \quad (456)$$

$$\text{EPOCH1} = \text{a keyword in the } \text{tgainfile}, \quad (457)$$

$$\text{EPOCH2} = \text{a keyword in the } \text{tgainfile}, \quad (458)$$

$$\delta_1 = \left(\frac{\text{PHA} - \text{PHA}_m[r]}{\text{PHA}_{m+1}[r] - \text{PHA}_m[r]} \right) (\text{DELTPHA1}_{m+1}[r] - \text{DELTPHA1}_m[r]) + \text{DELTPHA1}_m[r], \quad (459)$$

$$\text{DELTPHA1}_m[r], \quad (460)$$

$$\left\{ \begin{array}{l}
r \text{ is the row of the } \mathbf{tgainfile} \text{ where} \\
\left\{ \begin{array}{l}
\text{CCD_ID}[r] = \text{CCD_ID}, \\
\text{CHIPX_LO}[r] \leq \text{CHIPX}, \\
\text{CHIPX_HI}[r] \geq \text{CHIPX}, \\
\text{CHIPY_LO}[r] \leq \text{nint}(\text{CHIPY_ADJ}), \text{ and} \\
\text{CHIPY_HI}[r] \geq \text{nint}(\text{CHIPY_ADJ}).
\end{array} \right. \\
m \text{ is the element of row } r \text{ where} \\
\left\{ \begin{array}{l}
\text{PHA}_m[r] \leq \text{PHA} \text{ and} \\
\text{PHA}_{m+1}[r] > \text{PHA}. \\
\text{If } \text{PHA} < \text{PHA}_m[r] \text{ for } m = 0, \text{ then } m = 0. \\
\text{If } \text{PHA} \geq \text{PHA}_m[r] \text{ for } m = M \text{ and } M \text{ is the last element of } \text{PHA}[r], \\
\text{then } m = M - 1.
\end{array} \right. \\
\text{The } \mathbf{tgainfile} \text{ includes a binary table with columns named} \\
\text{CCD_ID, CHIPX_LO, CHIPX_HI, CHIPY_LO, CHIPY_HI, PHA, DELTPHA1, and} \\
\text{DELTPHA2.}
\end{array} \right. \quad (461)$$

$$\delta_2 = \left(\frac{\text{PHA} - \text{PHA}_m[r]}{\text{PHA}_{m+1}[r] - \text{PHA}_m[r]} \right) (\text{DELTPHA2}_{m+1}[r] - \text{DELTPHA2}_m[r]) + \quad (462)$$

$$\text{DELTPHA2}_m[r], \quad (463)$$

$$\epsilon = \text{is a uniform random deviate in the range } [0, 1), \quad (464)$$

$$\left\{ \begin{array}{l}
\text{If } \mathbf{rand_pha} = \text{no}, \text{ then } \epsilon = 0.
\end{array} \right. \quad (465)$$

(d) If

$$\text{PHA} \geq 32767, \quad (466)$$

then $\text{STATUS}[k] = 1$ for bit $k = 3$.

19. CORN_PHA:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (467)$$

then the value of CORN_PHA is read from the **infile**.

20. ENERGY:

(a) If the parameter **calculate_pi** = yes, if the parameter **gainfile** is specified, and if $\text{PHA} > 0$, then

i. The row i in the **gainfile** is identified such that

$$\text{CCD_ID} = \text{CCD_ID}_{\text{gain},i}, \quad (468)$$

$$\text{CHIPX_MIN}_{\text{gain},i} \leq \text{CHIPX} \leq \text{CHIPX_MAX}_{\text{gain},i}, \quad \text{and} \quad (469)$$

$$\text{CHIPY_MIN}_{\text{gain},i} \leq \text{nint}(\text{CHIPY_ADJ}) \leq \text{CHIPY_MAX}_{\text{gain},i}, \quad (470)$$

where $\text{CCD_ID}_{\text{gain}}$, $\text{CHIPX_MIN}_{\text{gain}}$, $\text{CHIPX_MAX}_{\text{gain}}$, $\text{CHIPY_MIN}_{\text{gain}}$, and $\text{CHIPY_MAX}_{\text{gain}}$ are columns in the **gainfile**.

ii. A uniform random deviate Δp is computed over the interval from $[-0.5, +0.5)$.

iii. The element j of row i of PHA_{gain} is identified such that

$$\text{PHA}_{\text{gain},i}[j] \leq (\text{PHA} + \Delta p) < \text{PHA}_{\text{gain},i}[j + 1], \quad (471)$$

where PHA_{gain} is a vector column in the **gainfile**. If $\text{PHA} + \Delta p < \text{PHA}_{\text{gain},i}[0]$, then $j = 0$. If $\text{PHA}_{\text{gain},i}[\text{NPOINTS} - 2] \leq \text{PHA} + \Delta p$, then $j = \text{NPOINTS} - 2$, where **NPOINTS** is a column in the **gainfile**.

iv. The ENERGY of an event is computed from the PHA of the event:

$$\text{ENERGY} = \left(\frac{\text{PHA} + \Delta p - \text{PHA}_{\text{gain},i}[j]}{\text{PHA}_{\text{gain},i}[j+1] - \text{PHA}_{\text{gain},i}[j]} \right) (\text{ENERGY}_{\text{gain},i}[j+1] - \text{ENERGY}_{\text{gain},i}[j]) + \text{ENERGY}_{\text{gain},i}[j], \quad (472)$$

where $\text{ENERGY}_{\text{gain}}$ is a vector column in the `gainfile`.

v. If $\text{ENERGY} < 0$, then $\text{ENERGY} = 0$.

- (b) If the parameter `calculate_pi` = yes, if the parameter `gainfile` is specified, and if $\text{PHA} \leq 0$, then $\text{ENERGY} = 0$.
- (c) If the parameter `calculate_pi` = no or if the parameter `gainfile` is not specified, then
- i. If the `infile` includes the ENERGY of an event, then the ENERGY of the event is equal to the ENERGY in the `infile`.
 - ii. If the `infile` does not include the ENERGY of an event, then $\text{ENERGY} = 0$.

21. PI:

(a) If
$$\text{calculate_pi} = \text{yes}, \quad (473)$$

then

i.

$$\text{PI} = \text{int} \left(\frac{\text{ENERGY}}{\text{pi_bin_width}} \right) + 1, \quad (474)$$

where “int” indicates the integer portion of what is in parentheses (i.e. the value is truncated or rounded down).

ii. If

$$\text{PI} < 1, \quad (475)$$

then $\text{PI} = 1$.

iii. If

$$\text{PI} > \text{pi_num_bins}, \quad (476)$$

then $\text{PI} = \text{pi_num_bins}$.

(b) If
$$\text{calculate_pi} = \text{no} \quad (477)$$

and if the `infile` includes the value of PI for an event, then the value of PI is read from the `infile`.

22. `pix_adj`:

(a) centroid:

If

$$\text{pix_adj} = \text{centroid}, \quad (478)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} - w'[0] + w'[2] - w'[3] + w'[5] - w'[6] + w'[8] \quad (479)$$

and if

$$\text{DATAMODE} = \text{FAINT} \text{ or} \quad (480)$$

$$\text{DATAMODE} = \text{FAINT_BIAS} \text{ or} \quad (481)$$

$$\text{DATAMODE} = \text{GRADED} \text{ or} \quad (482)$$

$$\text{DATAMODE} = \text{VF AINT}, \quad (483)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} - w'[0] - w'[1] - w'[2] + w'[6] + w'[7] + w'[8] \quad (484)$$

and if

$$\text{DATAMODE} = \text{CC33_FAINT} \text{ or} \quad (485)$$

$$\text{DATAMODE} = \text{CC33_GRADED}, \quad (486)$$

then

$$\text{TIME} = \text{TIME} + (w'[0] + w'[1] + w'[2] - w'[6] - w'[7] - w'[8]) \times \text{TIMEDEL}_{\text{in}}, \quad (487)$$

where

$$w'[j] = \frac{w[j]}{\sum_{j=0}^8 w[j]}, \quad (488)$$

$$w[j] = \begin{cases} p[j] & \text{if the pixel is valid} \\ 0 & \text{if the pixel is invalid,} \end{cases} \quad (489)$$

$$p[j] = \begin{cases} \text{PHAS_ADJ}[j] & \text{if apply_cti = yes} \\ \text{PHAS}[j] & \text{if apply_cti = no,} \end{cases} \quad (490)$$

and the pixel is invalid if

$$\beta[j] = 0 \text{ or} \quad (491)$$

$$\text{STATUS}[0] = 1 \text{ or} \quad (492)$$

$$\text{STATUS}[1] = 1 \text{ or} \quad (493)$$

$$\text{STATUS}[2] = 1 \text{ or} \quad (494)$$

$$\text{STATUS}[3] = 1 \text{ or} \quad (495)$$

$$\text{STATUS}[4] = 1 \text{ or} \quad (496)$$

$$\text{STATUS}[11] = 1 \text{ or} \quad (497)$$

$$\text{STATUS}[13] = 1 \text{ or} \quad (498)$$

$$\text{STATUS}[14] = 1 \text{ or} \quad (499)$$

$$\text{STATUS}[15] = 1 \text{ or} \quad (500)$$

$$\text{STATUS}[16] = 1. \quad (501)$$

Note that it is possible for the centroid algorithm to yield an adjustment to `CHIPX_ADJ` and/or `CHIPY_ADJ` that is greater than half a pixel. However, the adjustment cannot equal or exceed one pixel.

(b) edser:

If

$$\text{pix_adj} = \text{edser}, \quad (502)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} + \left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta X[k+1] - \Delta X[k]) + \Delta X[k] \quad (503)$$

and if

$$\text{DATAMODE} = \text{FAINT} \text{ or} \quad (504)$$

$$\text{DATAMODE} = \text{FAINT_BIAS} \text{ or} \quad (505)$$

$$\text{DATAMODE} = \text{GRADED} \text{ or} \quad (506)$$

$$\text{DATAMODE} = \text{VFAINT}, \quad (507)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k] \quad (508)$$

and if

$$\text{DATAMODE} = \text{CC33_FAINT or} \quad (509)$$

$$\text{DATAMODE} = \text{CC33_GRADED}, \quad (510)$$

then

$$\text{TIME} = \text{TIME} - \left(\left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k] \right) \times \text{TIMEDEL}_{\text{in}}, \quad (511)$$

where $E[k]$ and $E[k+1]$, $X[k]$ and $X[k+1]$, and $Y[k]$ and $Y[k+1]$ are the k and $(k+1)^{\text{th}}$ elements of the vector columns $\text{ENERGY}_{\text{subpix}}$, $\text{CHIPX_OFFSET}_{\text{subpix}}$, and $\text{CHIPY_OFFSET}_{\text{subpix}}$, respectively. These columns are in the HDU of the `subpixfile` where the value of the keyword `CCD_ID` is equal to the value of the `CCD_ID` of the event. The appropriate row of these columns is the one where $\text{FLTGRADE}_{\text{subpix}} = \text{FLTGRADE}$. The values of k are the ones where

$$\text{ENERGY} \geq E[k] \text{ and} \quad (512)$$

$$\text{ENERGY} < E[k+1]. \quad (513)$$

Note that if

$$\text{ENERGY} \leq E[0], \quad (514)$$

then $k = 0$. Similarly, if

$$\text{ENERGY} \geq E[\text{NPOINTS}_{\text{subpix}} - 2], \quad (515)$$

then $k = \text{NPOINTS}_{\text{subpix}} - 2$.

(c) none:

If

$$\text{pix_adj} = \text{none}, \quad (516)$$

then the values of `CHIPX_ADJ` and `CHIPY_ADJ` remain unchanged.

(d) randomize:

If

$$\text{pix_adj} = \text{randomize}, \quad (517)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} + \epsilon_x \quad (518)$$

and if

$$\text{DATAMODE} = \text{FAINT or} \quad (519)$$

$$\text{DATAMODE} = \text{FAINT_BIAS or} \quad (520)$$

$$\text{DATAMODE} = \text{GRADED or} \quad (521)$$

$$\text{DATAMODE} = \text{VF AINT}, \quad (522)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \epsilon_y \quad (523)$$

and if

$$\text{DATAMODE} = \text{CC33_FAINT or} \quad (524)$$

$$\text{DATAMODE} = \text{CC33_GRADED}, \quad (525)$$

then

$$\text{TIME} = \text{TIME} - \epsilon_y \times \text{TIMEDEL}_{\text{in}}, \quad (526)$$

where ϵ_x and ϵ_y are uniform random deviates in the range $[-0.5, +0.5)$ pixel.

(e) If

$$\text{CHIPX_ADJ} < 0.5, \quad (527)$$

then

$$\text{CHIPX_ADJ} = 1. \quad (528)$$

(f) If

$$\text{CHIPX_ADJ} \geq 1024.5, \quad (529)$$

then

$$\text{CHIPX_ADJ} = 1024. \quad (530)$$

(g) If

$$\text{CHIPY_ADJ} < 0.5, \quad (531)$$

then

$$\text{CHIPY_ADJ} = 1. \quad (532)$$

(h) If

$$\text{CHIPY_ADJ} \geq 1024.5, \quad (533)$$

then

$$\text{CHIPY_ADJ} = 1024. \quad (534)$$

23. TDET_X and TDET_Y:

(a) If

$$\text{stop} = \text{tdet or} \quad (535)$$

$$\text{stop} = \text{det or} \quad (536)$$

$$\text{stop} = \text{tan or} \quad (537)$$

$$\text{stop} = \text{sky}, \quad (538)$$

then the values of TDET_X and TDET_Y are computed using the values of $\text{nint}(\text{CHIPX_ADJ})$ and $\text{nint}(\text{CHIPY_ADJ})$. Here, “nint” indicates that the real-valued coordinate is rounded to the nearest integer before the computation of the TDET coordinate.

24. DET_X and DET_Y:

(a) If

`stop = det` or (539)

`stop = tan` or (540)

`stop = sky`, (541)

then the values of DETX and DETY are computed using the real-valued coordinates CHIPX_ADJ and CHIPY_ADJ.

25. X and Y:

(a) If

`stop = sky`, (542)

then the values of X and Y are computed using the real-valued coordinates CHIPX_ADJ and CHIPY_ADJ.

26. SKY_1D:

(a) If

`DATAMODEin = CC33_FAINT` or (543)

`DATAMODEin = CC33_GRADED` (544)

and if

`stop = sky`, (545)

then the value of SKY_1D is computed.

1.5.4 Write outfile

1. PIX_ADJ:

(a) If

`pix_adj = centroid`, (546)

then

`PIX_ADJ = CENTROID`, (547)

(b) If

`pix_adj = edser`, (548)

then

`PIX_ADJ = EDSER`, (549)

(c) If

`pix_adj = none`, (550)

then

`PIX_ADJ = NONE`, (551)

(d) If

$$\text{pix_adj} = \text{randomize}, \quad (552)$$

then

$$\text{PIX_ADJ} = \text{RANDOMIZE}, \quad (553)$$

2. RAND_SKY:

(a) If

$$\text{pix_adj} = \text{centroid}, \quad (554)$$

then

$$\text{RAND_SKY} = 0.0, \quad (555)$$

(b) If

$$\text{pix_adj} = \text{edser}, \quad (556)$$

then

$$\text{RAND_SKY} = 0.0, \quad (557)$$

(c) If

$$\text{pix_adj} = \text{none}, \quad (558)$$

then

$$\text{RAND_SKY} = 0.0, \quad (559)$$

(d) If

$$\text{pix_adj} = \text{randomize}, \quad (560)$$

then

$$\text{RAND_SKY} = 0.5, \quad (561)$$

3. TIME_ADJ:

(a) TE mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (562)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (563)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (564)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (565)$$

then

$$\text{TIME_ADJ} = \text{NONE}. \quad (566)$$

(b) Pointing CC mode without grating data:

i. If

$$\text{OBS_MODE} = \text{pointing or} \quad (567)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (568)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (569)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (570)$$

and

$$\text{CONTENT}_{\text{in}} = \text{EVT0 or} \quad (571)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (572)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (573)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (574)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3 \quad (575)$$

and

$$\cos(\text{DEC_ADJ}_I) \cos(\text{DEC_TARG}_{\text{in}}) \cos(\text{RA_ADJ}_I - \text{RA_TARG}_{\text{in}}) + \quad (576)$$

$$\sin(\text{DEC_ADJ}_I) \sin(\text{DEC_TARG}_{\text{in}}) < \quad (577)$$

$$4.855 \times 10^{-11}, \quad (578)$$

then

$$\text{TIME_ADJ} = \text{TARGET}. \quad (579)$$

ii. If

$$\text{OBS_MODE} = \text{pointing or} \quad (580)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (581)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (582)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (583)$$

and

$$\text{CONTENT}_{\text{in}} = \text{EVT0 or} \quad (584)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (585)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (586)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (587)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9 \quad (588)$$

and

$$\cos(\text{DEC_ADJ}_S) \cos(\text{DEC_TARG}_{in}) \cos(\text{RA_ADJ}_S - \text{RA_TARG}_{in}) + \quad (589)$$

$$\sin(\text{DEC_ADJ}_S) \sin(\text{DEC_TARG}_{in}) < \quad (590)$$

$$4.855 \times 10^{-11}, \quad (591)$$

then

$$\text{TIME_ADJ} = \text{TARGET}. \quad (592)$$

iii. If

$$\text{OBS_MODE} = \text{pointing or} \quad (593)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (594)$$

and

$$\text{DATAMODE}_{in} = \text{CC33_FAINT or} \quad (595)$$

$$\text{DATAMODE}_{in} = \text{CC33_GRADED} \quad (596)$$

and

$$\text{CONTENT}_{in} = \text{EVT0 or} \quad (597)$$

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (598)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (599)$$

and

$$\text{CCD_ID}_{focus} \geq 0 \text{ and} \quad (600)$$

$$\text{CCD_ID}_{focus} \leq 3 \quad (601)$$

and

$$\cos(\text{DEC_ADJ}_I) \cos(\text{DEC_TARG}_{in}) \cos(\text{RA_ADJ}_I - \text{RA_TARG}_{in}) + \quad (602)$$

$$\sin(\text{DEC_ADJ}_I) \sin(\text{DEC_TARG}_{in}) \geq \quad (603)$$

$$4.855 \times 10^{-11}, \quad (604)$$

then

$$\text{TIME_ADJ} = \text{AIMPOINT}. \quad (605)$$

iv. If

$$\text{OBS_MODE} = \text{pointing or} \quad (606)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (607)$$

and

$$\text{DATAMODE}_{in} = \text{CC33_FAINT or} \quad (608)$$

$$\text{DATAMODE}_{in} = \text{CC33_GRADED} \quad (609)$$

and

$$\text{CONTENT}_{in} = \text{EVT0 or} \quad (610)$$

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (611)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (612)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (613)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9 \quad (614)$$

and

$$\cos(\text{DEC_ADJ}_S) \cos(\text{DEC_TARG}_{\text{in}}) \cos(\text{RA_ADJ}_S - \text{RA_TARG}_{\text{in}}) + \quad (615)$$

$$\sin(\text{DEC_ADJ}_S) \sin(\text{DEC_TARG}_{\text{in}}) \geq \quad (616)$$

$$4.855 \times 10^{-11}, \quad (617)$$

then

$$\text{TIME_ADJ} = \text{AIMPOINT}. \quad (618)$$

(c) Pointing CC mode with ACIS-S grating data:

If

$$\text{OBS_MODE} = \text{pointing or} \quad (619)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (620)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (621)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (622)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (623)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (624)$$

$$\text{CCD_ID}_{\text{focus}} \leq 9, \quad (625)$$

then

$$\text{TIME_ADJ} = \text{GRATING}. \quad (626)$$

(d) Pointing CC mode with ACIS-I grating data:

i. If

$$\text{OBS_MODE} = \text{pointing or} \quad (627)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (628)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (629)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (630)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (631)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (632)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3 \quad (633)$$

and

$$\cos(\text{DEC_ADJ}_I) \cos(\text{DEC_TARG}_{\text{in}}) \cos(\text{RA_ADJ}_I - \text{RA_TARG}_{\text{in}}) + \quad (634)$$

$$\sin(\text{DEC_ADJ}_I) \sin(\text{DEC_TARG}_{\text{in}}) < \quad (635)$$

$$4.855 \times 10^{-11}, \quad (636)$$

then

$$\text{TIME_ADJ} = \text{TARGET}. \quad (637)$$

ii. If

$$\text{OBS_MODE} = \text{pointing or} \quad (638)$$

$$\text{OBS_MODE} = \text{POINTING} \quad (639)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (640)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (641)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (642)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 0 \text{ and} \quad (643)$$

$$\text{CCD_ID}_{\text{focus}} \leq 3 \quad (644)$$

and

$$\cos(\text{DEC_ADJ}_I) \cos(\text{DEC_TARG}_{\text{in}}) \cos(\text{RA_ADJ}_I - \text{RA_TARG}_{\text{in}}) + \quad (645)$$

$$\sin(\text{DEC_ADJ}_I) \sin(\text{DEC_TARG}_{\text{in}}) \geq \quad (646)$$

$$4.855 \times 10^{-11}, \quad (647)$$

then

$$\text{TIME_ADJ} = \text{AIMPOINT}. \quad (648)$$

(e) Secondary CC mode:

If

$$\text{OBS_MODE} \neq \text{pointing and} \quad (649)$$

$$\text{OBS_MODE} \neq \text{POINTING} \quad (650)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (651)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (652)$$

then

$$\text{TIME_ADJ} = \text{MIDCHIP}. \quad (653)$$

2 TBD

- Complete the spec to include all of the timed exposure mode processing.
- Complete sections 1.1, 1.2, 1.3, and 1.4.
- Should CONTENTs other than EVT0, EVT1, TGEVT1, and EVT2 be included?
- Should CONTENT = EVT2 be dropped?
- Should DATAMODEs other than CC33_FAINT, CC33_GRADED, FAINT, FAINT_BIAS, GRADED, and VFaint be included?
- Are the RA_TARG, DEC_TARG, RA_NOM, DEC_NOM, and TIMEDEL keywords in the output of afe (need obsfile sometimes)?
- What if TIME_RO is not in the infile (output of afe? EVT2 files)?
- What if a small fraction of the values of CHIPY_TARG are off the chip due to bad aspect?
- Make sure that the STATUS bits are unset and reset properly.
- What about aoff and soff files instead of asol files?
- Are the β in PHA_RO the same as the β in PHA?
- Should something be done about SKY_1D?