



MIT Kavli Institute



Chandra X-Ray Center

## MEMORANDUM

August 4, 2015

**To:** Jonathan McDowell, SDS Group Leader  
**From:** Glenn E. Allen, SDS  
**Subject:** `acis_process_events` spec  
**Revision:** 3.34  
**URL:** <http://space.mit.edu/CXC/docs/docs.html#ape>  
**File:** `/nfs/inconceivable/d0/sds/specs/acis_process_events/ape_spec_3.34.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

`obsfile`  $\neq$  none and (3)

`obsfile`  $\neq$  NONE (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

`obsfile`  $\neq$  none and (5)

`obsfile`  $\neq$  NONE, (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

`obs_mode` = pointing or (7)

`obs_mode` = POINTING or (8)

`obs_mode` = secondary or (9)

`obs_mode` = SECONDARY, (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

`obs_mode`  $\neq$  pointing and (11)

`obs_mode`  $\neq$  POINTING and (12)

`obs_mode`  $\neq$  secondary and (13)

`obs_mode`  $\neq$  SECONDARY, (14)

then `OBS_MODE` is set to “none”.

2. `acaofffile`:

(a) Validation:

If

`OBS_MODE` = pointing or (15)

`OBS_MODE` = POINTING, (16)

then

i. Setting:

If

`acaofffile` = none or (17)

`acaofffile` = NONE, (18)

then `acis_process_events` produces a warning message.

ii. Existence:

If

$$\text{acaofffile} \neq \text{none} \text{ and} \tag{19}$$

$$\text{acaofffile} \neq \text{NONE} \tag{20}$$

and the `acaofffile` does not exist, then `acis_process_events` produces a warning and `acaofffile` is set to “none.”

iii. Permission:

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

iv. CONTENT:

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

$$\text{CONTENT} = \text{ASPSOL}, \tag{21}$$

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

v. Keyword:

If HDU  $h_{\text{acaoff}}$  of the `acaofffile` does not include the keyword `TSTART`, then `acis_process_events` produces a warning message and `acaofffile` is set to “none.”

vi. Columns:

If HDU  $h_{\text{acaoff}}$  of the `acaofffile` does not include the columns `TIME`, `RA`, `DEC`, and `ROLL` then `acis_process_events` produces a warning message and `acaofffile` is set to “none.” 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 and the the values `TSTART` are not in increasing order, then `acis_process_events` produces a warning message and `acaofffile` is set to “none.”

### 3. `alignmentfile`:

(a) Validation:

If

$$\text{OBS\_MODE} = \text{pointing} \text{ or} \tag{22}$$

$$\text{OBS\_MODE} = \text{POINTING}, \tag{23}$$

then

i. Setting:

If

$$\text{alignmentfile} = \text{none} \text{ or} \tag{24}$$

$$\text{alignmentfile} = \text{NONE}, \tag{25}$$

then `acis_process_events` produces a warning message.

ii. Existence:

If

$$\text{alignmentfile} \neq \text{none} \text{ and} \tag{26}$$

$$\text{alignmentfile} \neq \text{NONE} \tag{27}$$

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

iii. Permission:

If the `alignmentfile` exists and the file permissions do not allow it to be read, then `acis-process_events` produces a warning message and `alignmentfile` is set to “none.”.

iv. CONTENT:

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

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

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

v. Keyword:

If HDU  $h_{\text{alignment}}$  of the `alignmentfile` does not include the keyword `TSTART`, then `acis-process_events` produces a warning message and `alignmentfile` is set to “none.”.

vi. Columns:

If HDU  $h_{\text{alignment}}$  of the `alignmentfile` does not include the columns `DY`, `DZ`, and `DTHETA` then `acis_process_events` produces a warning message and `alignmentfile` is set to “none.”.

vii. Sequential:

If more than one valid `alignmentfile` is specified and the values `TSTART` are not in increasing order, then `acis_process_events` produces a warning message and `alignmentfile` is set to “none.”.

#### 4. `fltfile`:

(a) Validation:

i. Existence:

If

$$\text{fltfile} \neq \text{none} \text{ and} \quad (29)$$

$$\text{fltfile} \neq \text{NONE} \quad (30)$$

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

ii. Permission:

If the `fltfile` exists and the file permissions do not allow it to be read, then `acis-process_events` produces a warning message and `fltfile` is set to “none.”.

iii. CONTENT:

If the `fltfile` does not have one or more HDUs  $h_{\text{flt}}$  with the keyword

$$\text{CONTENT} = \text{GTI}, \quad (31)$$

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

iv. Columns:

If the HDUs  $h_{\text{flt}}$  of the `fltfile` do not include the columns `START` and `STOP`, then `acis-process_events` produces a warning message and `fltfile` is set to “none.”.

#### 5. `infile`:

(a) Existence:

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

(b) Permission:

If the `infile` exists and 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  $h_{in}$  keyword of the same name. Hereafter this keyword is referred to as OBS\_MODE.
- B. If the HDU  $h_{in}$  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

$$\text{obs\_mode} \neq \text{pointing and} \quad (32)$$

$$\text{obs\_mode} \neq \text{POINTING and} \quad (33)$$

$$\text{obs\_mode} \neq \text{secondary and} \quad (34)$$

$$\text{obs\_mode} \neq \text{SECONDARY,} \quad (35)$$

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

$$\text{DATAMODE} \neq \text{CC33\_FAINT and} \quad (36)$$

$$\text{DATAMODE} \neq \text{CC33\_GRADED and} \quad (37)$$

$$\text{DATAMODE} \neq \text{FAINT and} \quad (38)$$

$$\text{DATAMODE} \neq \text{FAINT\_BIAS and} \quad (39)$$

$$\text{DATAMODE} \neq \text{GRADED and} \quad (40)$$

$$\text{DATAMODE} \neq \text{VFAINT,} \quad (41)$$

then `acis_process_events` exits with an error message. Hereafter, the value of this keyword is referred to as  $\text{DATAMODE}_{in}$ .

iii. CONTENT:

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

$$\text{CONTENT} = \text{EVT0 or} \quad (42)$$

$$\text{CONTENT} = \text{EVT1 or} \quad (43)$$

$$\text{CONTENT} = \text{TGEVT1 or} \quad (44)$$

$$\text{CONTENT} = \text{EVT2,} \quad (45)$$

then `acis_process_events` exits with an error message. Hereafter, the value of this keyword is referred to as  $\text{CONTENT}_{in}$ .

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  $\text{TIME}_{in}$ .

v. TIME\_R0:

If

$$\text{DATAMODE}_{in} = \text{CC33\_FAINT or} \quad (46)$$

$$\text{DATAMODE}_{in} = \text{CC33\_GRADED} \quad (47)$$

and

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (48)$$

$$\text{CONTENT}_{in} = \text{TGEVT1 or} \quad (49)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (50)$$

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

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

$$\text{CONTENT}_{in} = \text{EVT0} \quad (51)$$

and 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

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (52)$$

$$\text{CONTENT}_{in} = \text{TGEVT1 or} \quad (53)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (54)$$

and 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

$$\text{CONTENT}_{in} = \text{EVT0} \quad (55)$$

and 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

$$\text{CONTENT}_{in} = \text{EVT1 or} \quad (56)$$

$$\text{CONTENT}_{in} = \text{TGEVT1 or} \quad (57)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (58)$$

and 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

$$\text{CONTENT}_{in} = \text{EVT0} \quad (59)$$

and

$$\text{DATAMODE}_{in} = \text{FAINT or} \quad (60)$$

$$\text{DATAMODE}_{in} = \text{FAINT\_BIAS or} \quad (61)$$

$$\text{DATAMODE}_{in} = \text{GRADED or} \quad (62)$$

$$\text{DATAMODE}_{in} = \text{VFAINT} \quad (63)$$

and 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

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (64)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ or} \quad (65)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (66)$$

and HDU  $h_{\text{in}}$  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

$$\text{CONTENT}_{\text{in}} = \text{EVT1} \text{ or} \quad (67)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ or} \quad (68)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (69)$$

and HDU  $h_{\text{in}}$  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

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ or} \quad (70)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (71)$$

and HDU  $h_{\text{in}}$  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_ZO`, and `TG_M`:

If

$$\text{OBS\_MODE} = \text{pointing} \text{ or} \quad (72)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (73)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ or} \quad (74)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (75)$$

then

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

If HDU  $h_{\text{in}}$  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_ZO`, and `TG_M`:

If

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (76)$$

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

6. `stop`:

(a) Lowercase:

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

(b) Validation:

i. Setting:

If

`stop`  $\neq$  none and (77)

`stop`  $\neq$  chip and (78)

`stop`  $\neq$  tdet and (79)

`stop`  $\neq$  det and (80)

`stop`  $\neq$  tan and (81)

`stop`  $\neq$  sky, (82)

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

ii. `OBS_MODE`:

If

`OBS_MODE`  $\neq$  pointing and (83)

`OBS_MODE`  $\neq$  POINTING (84)

and

`stop`  $\neq$  none and (85)

`stop`  $\neq$  chip and (86)

`stop`  $\neq$  tdet, (87)

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

iii. `acaofffile`:

If

`OBS_MODE` = pointing or (88)

`OBS_MODE` = POINTING (89)

and

`acaofffile` = none and (90)

`stop`  $\neq$  none and (91)

`stop`  $\neq$  chip and (92)

`stop`  $\neq$  tdet, (93)

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

iv. `alignmentfile`:

If

`OBS_MODE` = pointing or (94)

`OBS_MODE` = POINTING (95)



and

`alignmentfile = none and` (96)

`stop ≠ none and` (97)

`stop ≠ chip and` (98)

`stop ≠ tdet,` (99)

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

7. `doevtgrade`:

(a) Lowercase:

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

(b) Validation:

If

`doevtgrade ≠ yes and` (100)

`doevtgrade ≠ no,` (101)

then `acis_process_events` exits with an error message.

8. `apply_cti`:

(a) Lowercase:

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

(b) Validation:

i. Setting:

If

`apply_cti ≠ yes and` (102)

`apply_cti ≠ no,` (103)

then `acis_process_events` exits with an error message.

ii. PHAS:

If

`apply_cti = yes` (104)

and 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

`apply_cti = yes and` (105)

`doevtgrade = no,` (106)

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

9. `badpixfile`:

(a) Validation:

i. Existence:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (107)$$

$$\text{badpixfile} \neq \text{NONE} \quad (108)$$

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

ii. Permission:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (109)$$

$$\text{badpixfile} \neq \text{NONE} \quad (110)$$

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

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (111)$$

$$\text{badpixfile} \neq \text{NONE} \quad (112)$$

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

$$\text{CONTENT} = \text{BADPIX} \text{ or} \quad (113)$$

$$\text{CONTENT} = \text{CDB\_ACIS\_BADPIX}, \quad (114)$$

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

iv. Keyword:

If

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (115)$$

$$\text{badpixfile} \neq \text{NONE} \quad (116)$$

and the HDU(s)  $h_{\text{badpix}}$  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

$$\text{badpixfile} \neq \text{none} \text{ and} \quad (117)$$

$$\text{badpixfile} \neq \text{NONE} \quad (118)$$

and the HDU(s)  $h_{\text{badpix}}$  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.

10. `ctifile`:

(a) Validation:

If

$$\text{ctifile} \neq \text{caldb} \text{ and} \tag{119}$$

$$\text{ctifile} \neq \text{CALDB}, \tag{120}$$

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 and 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  $h_{\text{cti}}$  with the keyword

$$\text{CONTENT} = \text{CDB\_ACIS\_CTI}, \tag{121}$$

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.

11. `clobber`:

(a) Lowercase:

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

(b) Validation:

i. Setting:

If

$$\text{clobber} \neq \text{yes} \text{ and} \tag{122}$$

$$\text{clobber} \neq \text{no}, \tag{123}$$

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

ii. Permission:

If

$$\text{clobber} = \text{yes} \tag{124}$$

and the `outfile` exists and 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

$$\text{clobber} = \text{no} \tag{125}$$

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

12. `pix_adj`:

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

(b) Validation:

i. Setting:  
If

`pix_adj`  $\neq$  centroid and (126)

`pix_adj`  $\neq$  edser and (127)

`pix_adj`  $\neq$  none and (128)

`pix_adj`  $\neq$  randomize, (129)

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

ii. OBS\_MODE:

If

OBS\_MODE  $\neq$  pointing and (130)

OBS\_MODE  $\neq$  POINTING (131)

and

`pix_adj`  $\neq$  none, (132)

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

iii. stop:

If

`pix_adj` = centroid or (133)

`pix_adj` = edser or (134)

`pix_adj` = randomize (135)

and

`stop`  $\neq$  sky, (136)

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

iv. PHAS:

If

`pix_adj` = centroid (137)

and 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

`pix_adj` = edser (138)

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

13. `subpixfile`:

(a) If

$$\text{pix\_adj} = \text{edser}, \quad (139)$$

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 and 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  $h_{\text{subpix}}$  with the keyword

$$\text{CONTENT} = \text{AXAF\_SUBPIX}, \quad (140)$$

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

B. Keyword:

If the HDUs  $h_{\text{subpix}}$  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  $h_{\text{subpix}}$  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

$$\text{OBS\_MODE} = \text{pointing or} \quad (141)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (142)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (143)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (144)$$

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. `acaofffile`:

If

$$\text{OBS\_MODE} = \text{pointing or} \quad (145)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (146)$$

---

\*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.

and

$$\text{acaofffile} \neq \text{none and} \quad (147)$$

$$\text{alignmentfile} \neq \text{none}, \quad (148)$$

then

- (a)  $\text{TIME}_{\min}$  and  $\text{TIME}_{\max}$ :

The `acaofffile` data are processed to determine the earliest and latest times for which there is aspect information:

$$\text{TIME}_{\min} = \min(\text{TIME}_{\text{acaoff}}) \text{ and} \quad (149)$$

$$\text{TIME}_{\max} = \max(\text{TIME}_{\text{acaoff}}). \quad (150)$$

- (b)  $\text{CHIPY\_TARG}_{\text{med}}$ :

The values of  $\text{CHIPY\_TARG}_{\text{acaoff}}$  are computed from the values of `RA_TARG` and `DEC_TARG` using the orientation of the telescope (i.e. `RA`, `DEC`, and `ROLL`) and the orientation of the SIM (i.e. `DY`, `DZ`, and `DTHETA`) and the `TIMES` in the `acaofffile`. The values of  $\text{CHIPY\_TARG}_{\text{acaoff}}$  are processed to obtain the median value:

$$\text{CHIPY\_TARG}_{\text{med}} = \text{median}(\text{CHIPY\_TARG}_{\text{asol}}). \quad (151)$$

### 1.5.3 Loop over events

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

#### 1. STATUS:

- (a) Exists:

If HDU  $h_{\text{in}}$  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

$$\text{doevtgrade} = \text{yes}, \quad (152)$$

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  $h_{\text{in}}$  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  $\text{EXPNO}_{\text{in}}$ .

- (b) Validation:

If

$$\text{EXPNO} < 0 \text{ or} \quad (153)$$

$$\text{EXPNO} \geq 10^8, \quad (154)$$

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

$$\text{CCD\_ID} < 0 \text{ or} \tag{155}$$

$$\text{CCD\_ID} > 9, \tag{156}$$

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 and CHIPX\_ADJ:

(a) Read:

i. Level 0:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \tag{157}$$

then the value of CHIPX for an event is given by

$$\text{CHIPX} = \text{CCDX}_{\text{in}} + 1. \tag{158}$$

ii. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \tag{159}$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \tag{160}$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2}, \tag{161}$$

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

iii. CHIPX\_ADJ:

$$\text{CHIPX\_ADJ} = \text{CHIPX}. \tag{162}$$

(b) Validation:

i. Unphysical:

If

$$\text{CHIPX} < 1 \text{ or} \tag{163}$$

$$\text{CHIPX} > 1024, \tag{164}$$

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

$$\text{CHIPX} = 1 \text{ or} \tag{165}$$

$$\text{CHIPX} = 1024, \tag{166}$$

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

$$\text{NODE\_ID} = \text{int} \left( \frac{\text{CHIPX} - 1}{256} \right), \quad (167)$$

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

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (168)$$

then

A. Timed-exposure mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (169)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT\_BIAS or} \quad (170)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (171)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (172)$$

then the value of CHIPY for an event is given by

$$\text{CHIPY} = \text{CCDY}_{\text{in}} + 1. \quad (173)$$

B. Continuous-clocking mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (174)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (175)$$

then the value of CHIPY for an event is given by

$$\text{CHIPY} = \text{TROW}_{\text{in}} + 1. \quad (176)$$

ii. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (177)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (178)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2}, \quad (179)$$

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

(b) Validation:

i. Unphysical:



A. Timed-exposure mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (180)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT\_BIAS or} \quad (181)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (182)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \quad (183)$$

and

$$\text{CHIPY} < 1 \text{ or} \quad (184)$$

$$\text{CHIPY} > 1024, \quad (185)$$

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

B. Continuous-clocking mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (186)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (187)$$

and

$$\text{CHIPY} < 1 \text{ or} \quad (188)$$

$$\text{CHIPY} > 512, \quad (189)$$

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. FAIN, FAIN, or GRADED:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (190)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT\_BIAS or} \quad (191)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED} \quad (192)$$

and

$$\text{CHIPY} = 1 \text{ or} \quad (193)$$

$$\text{CHIPY} = 1024, \quad (194)$$

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

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \quad (195)$$

and

$$\text{CHIPY} = 1 \text{ or} \quad (196)$$

$$\text{CHIPY} = 2 \text{ or} \quad (197)$$

$$\text{CHIPY} = 1023 \text{ or} \quad (198)$$

$$\text{CHIPY} = 1024, \quad (199)$$

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

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (200)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (201)$$

and

$$\text{CHIPY} = 1 \text{ or} \quad (202)$$

$$\text{CHIPY} = 512, \quad (203)$$

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. `CHIPY_TARG`:

(a) Calculate:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (204)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (205)$$

and

$$\text{acaofffile} \neq \text{none and} \quad (206)$$

$$\text{acaofffile} \neq \text{NONE}, \quad (207)$$

then the value of `CHIPY_TARG` is computed from the values of `RA_TARG` and `DEC_TARG` using the orientation of the telescope (i.e. `RA`, `DEC`, and `ROLL`) and the orientation of the SIM (i.e. `DY`, `DZ`, and `DTHETA`) at the time `TIME`. The value of `CHIPY_TARG` can be negative. The coordinate `CHIPY_TARG` is used to compute the values of `X`, `Y`, and `SKY_1D`.

(b) Validation:

If

$$\text{CHIPY\_TARG} < -256 \text{ or} \quad (208)$$

$$\text{CHIPY\_TARG} \geq 1280, \quad (209)$$

then `acis_process_events` produces a warning message.

8. `TG_M`:

(a) Continuous-clocking mode with gratings:

If

$$\text{OBS\_MODE} = \text{pointing or} \quad (210)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (211)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (212)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (213)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (214)$$

then

i. Read:

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

ii. Validation:

A. If

$$\text{TG\_M} < -99, \quad (215)$$

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.

B. If

$$\text{TG\_M} > 99, \quad (217)$$

then

$$\text{TG\_M} = 99 \quad (218)$$

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.

## 9. CHIPY\_TG:

(a) Continuous-clocking mode with gratings:

If

$$\text{OBS\_MODE} = \text{pointing or} \quad (219)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (220)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (221)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (222)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (223)$$

then

i. Read:

The value of `CHIPY_TG` for an event is given by `CHIPY_TGin`.

ii. Validation:

A. If

$$\text{TG\_M} > -99 \text{ and} \quad (224)$$

$$\text{TG\_M} < 99 \text{ and} \quad (225)$$

$$\text{CHIPY\_TG} \neq \text{NULL} \quad (226)$$

and

$$\text{CHIPY\_TG} \leq 0 \text{ or} \quad (227)$$

$$\text{CHIPY\_TG} \geq 1025, \quad (228)$$

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

$$\text{TG\_M} > -99 \text{ and} \quad (229)$$

$$\text{TG\_M} < 99 \text{ and} \quad (230)$$

$$\text{CHIPY\_TG} \neq \text{NULL} \text{ and} \quad (231)$$

$$\text{CHIPY\_TG} < 1, \quad (232)$$

then

$$\text{CHIPY\_TG} = 1. \quad (233)$$

C. If

$$\text{TG\_M} > -99 \text{ and} \quad (234)$$

$$\text{TG\_M} < 99 \text{ and} \quad (235)$$

$$\text{CHIPY\_TG} \neq \text{NULL} \text{ and} \quad (236)$$

$$\text{CHIPY\_TG} > 1024, \quad (237)$$

then

$$\text{CHIPY\_TG} = 1024. \quad (238)$$

#### 10. `CHIPY_Z0`:

(a) Continuous-clocking mode with gratings:

If

$$\text{OBS\_MODE} = \text{pointing or} \quad (239)$$

$$\text{OBS\_MODE} = \text{POINTING} \quad (240)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (241)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (242)$$

and

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

then

i. Read:

The value of `CHIPY_Z0` for an event is given by `CHIPY_Z0in`.

11. `TIME_RO`:

(a) Continuous-clocking mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ or} \quad (244)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (245)$$

then

i. Read:

A. Level 0:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (246)$$

then the value of `TIME_RO` for an event is given by `TIMEin`.

B. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1} \text{ or} \quad (247)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ or} \quad (248)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (249)$$

and

$$\text{TIME\_RO}_{\text{in}} > 0, \quad (250)$$

then

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

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1} \text{ or} \quad (252)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ or} \quad (253)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (254)$$

and

$$\text{TIME\_RO}_{\text{in}} = 0, \quad (255)$$

then

$$\text{TIME\_RO} = \text{TIME}_{\text{in}}. \quad (256)$$

ii. Validation:

If

$$\text{TIME\_RO} < 0 \text{ or} \quad (257)$$

$$\text{TIME\_RO} \geq 3 \times 10^9, \quad (258)$$

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.

12. TIME and CHIPY\_ADJ:

(a) Read or calculate:

i. Timed exposure mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (259)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT\_BIAS or} \quad (260)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (261)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT,} \quad (262)$$

then

$$\text{TIME} = \text{TIME}_{\text{in}} \text{ and} \quad (263)$$

$$\text{CHIPY\_ADJ} = \text{CHIPY.} \quad (264)$$

ii. Continuous-clocking mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT or} \quad (265)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED} \quad (266)$$

then

A. Set

$$\text{CHIPY\_ADJ} = 512 \text{ and} \quad (267)$$

$$\text{TIME}' = \text{TIME}_{\text{in}}. \quad (268)$$

B. If

$$\text{acaofffile} \neq \text{none and} \quad (269)$$

$$\text{acaofffile} \neq \text{NONE,} \quad (270)$$

then

$$\text{CHIPY\_ADJ}' = \text{CHIPY\_TARG and} \quad (271)$$

$$\text{TIME}' = \text{TIME\_RO} - (\text{CHIPY\_ADJ}' + 1028) \times \text{TIMEDEL}, \quad (272)$$

and CHIPY\_ADJ is computed from the values of RA\_TARG and DEC\_TARG using the orientation of the telescope (i.e. RA, DEC, and ROLL) and the orientation of the SIM (i.e. DY, DZ, and DTHETA) at the time TIME'. At this step, the value of CHIPY\_ADJ can be negative.

C. If

$$\text{CONTENT} = \text{TGEVT1 and} \quad (273)$$

$$\text{TG\_M} > -99 \text{ and} \quad (274)$$

$$\text{TG\_M} < 99 \text{ and} \quad (275)$$

$$\text{CHIPY\_TG} \neq \text{NULL,} \quad (276)$$

then

$$\text{CHIPY\_ADJ} = \text{CHIPY\_TG.} \quad (277)$$

D. If

$$\text{CONTENT} = \text{TGEVT} \quad (278)$$

and

$$\text{TG\_M} = -99 \text{ or} \quad (279)$$

$$\text{TG\_M} = 99, \quad (280)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (281)$$

E. If

$$\text{acaofffile} \neq \text{none and} \quad (282)$$

$$\text{acaofffile} \neq \text{NONE} \quad (283)$$

and

$$\text{TIME}' < \text{TIME}_{\text{min}} \text{ or} \quad (284)$$

$$\text{TIME}' \geq \text{TIME}_{\text{max}}, \quad (285)$$

then

$$\text{CHIPY\_ADJ} = \text{CHIPY\_TARG}_{\text{med}}. \quad (286)$$

F. If

$$\text{acaofffile} = \text{none or} \quad (287)$$

$$\text{acaofffile} = \text{NONE} \quad (288)$$

and

$$\text{CONTENT} \neq \text{TGEVT1}, \quad (289)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (290)$$

G. If

$$\text{CCD\_ID}_{<\text{focus}} \leq 3 \text{ and} \quad (291)$$

$$\text{CCD\_ID} \geq 4, \quad (292)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (293)$$

H. If

$$\text{CCD\_ID}_{\text{focus}} \geq 4 \text{ and} \quad (294)$$

$$\text{CCD\_ID} \leq 3, \quad (295)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (296)$$

I. If

$$\text{obs\_mode} = \text{secondary or} \quad (297)$$

$$\text{obs\_mode} = \text{SECONDARY}, \quad (298)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (299)$$

J. If

$$\text{CHIPY\_ADJ} < 0.5 \text{ or} \quad (300)$$

$$\text{CHIPY\_ADJ} \geq 1024.5 \quad (301)$$

then

$$\text{CHIPY\_ADJ} = 512. \quad (302)$$

K. Set

$$\text{TIME} = \text{TIME\_RO} - (\text{CHIPY\_ADJ} + 1028) \times \text{TIMEDEL}. \quad (303)$$

In continuous-clocking mode, the coordinate `CHIPY_ADJ` is used to compute the time, the pulse heights, and the coordinates (except for `X`, `Y`, and `SKY_1D`).

(b) Validation:

i. If

$$\text{TIME} < 0 \text{ or} \quad (304)$$

$$\text{TIME} \geq 3 \times 10^9, \quad (305)$$

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

$$\text{CHIPY\_ADJ} < 0.5 \text{ or} \quad (306)$$

$$\text{CHIPY\_ADJ} \geq 1024.5, \quad (307)$$

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

13. Bad pixel:

(a) If

$$\text{badpixfile} \neq \text{none and} \quad (308)$$

$$\text{badpixfile} \neq \text{NONE} \quad (309)$$

and the `badpixfile` includes a valid HDU  $h_{\text{badpix}}$  where  $\text{CCD\_ID}_{\text{badpix}} = \text{CCD\_ID}$ , then the HDU  $h_{\text{badpix}}$  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 there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (310)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (311)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (312)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (313)$$

and

$$\text{STATUS}_{\text{badpix},r}[5] = 1 \text{ or} \quad (314)$$

$$\text{STATUS}_{\text{badpix},r}[6] = 1 \text{ or} \quad (315)$$

$$\text{STATUS}_{\text{badpix},r}[9] = 1, \quad (316)$$

then

$$\text{STATUS}[0] = 1 \quad (317)$$

for the event. Here  $\text{CCD\_ID}_{\text{badpix}}$  is the value of the keyword  $\text{CCD\_ID}$  in HDU  $h_{\text{badpix}}$  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  $h_{\text{badpix}}$  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  $h_{\text{badpix}}$  of the `badpixfile`.

- ii. If  $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$  or  $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$  and there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (318)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (319)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (320)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (321)$$

and

$$\text{STATUS}_{\text{badpix},r}[0] = 1 \text{ or} \quad (322)$$

$$\text{STATUS}_{\text{badpix},r}[1] = 1 \text{ or} \quad (323)$$

$$\text{STATUS}_{\text{badpix},r}[7] = 1 \text{ or} \quad (324)$$

$$\text{STATUS}_{\text{badpix},r}[11] = 1 \text{ or} \quad (325)$$

$$\text{STATUS}_{\text{badpix},r}[12] = 1 \text{ or} \quad (326)$$

$$\text{STATUS}_{\text{badpix},r}[13] = 1 \text{ or} \quad (327)$$

$$\text{STATUS}_{\text{badpix},r}[14] = 1 \text{ or} \quad (328)$$

$$\text{STATUS}_{\text{badpix},r}[16] = 1, \quad (329)$$

then

$$\text{STATUS}[4] = 1 \quad (330)$$

for the event.

- iii. If  $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$  or  $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$  and there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (331)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (332)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (333)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (334)$$

and

$$\text{STATUS}_{\text{badpix},r}[8] = 1 \text{ or} \quad (335)$$

$$\text{STATUS}_{\text{badpix},r}[10] = 1, \quad (336)$$

then

$$\text{STATUS}[5] = 1 \quad (337)$$

for the event.

- iv. If  $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$  or  $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$  and there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (338)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (339)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (340)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (341)$$

and

$$\text{STATUS}_{\text{badpix},r}[3] = 1, \quad (342)$$

then

$$\text{STATUS}[6] = 1 \quad (343)$$

for the event.

- v. If  $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$  or  $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$  and there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (344)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (345)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (346)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (347)$$

and

$$\text{STATUS}_{\text{badpix},r}[2] = 1 \text{ or} \quad (348)$$

$$\text{STATUS}_{\text{badpix},r}[4] = 1, \quad (349)$$

then

$$\text{STATUS}[8] = 1 \quad (350)$$

for the event.

- vi. If  $\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}$  or  $\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}$  and there are one or more rows  $r$  in HDU  $h_{\text{badpix}}$  where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (351)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (352)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (353)$$

$$\text{TIME} < \text{TIME\_STOP}_{\text{badpix},r} \quad (354)$$

and

$$\text{STATUS}_{\text{badpix},r}[15] = 1, \quad (355)$$

then

$$\text{STATUS}[16] = 1 \quad (356)$$

for the event.

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

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

14. 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  $k = 2$ .

15. PHAS\_ADJ:

- (a) If HDU 1 of the `infile` includes `DATAMODEin = CC33_FAINT` and the parameter `apply_cti = yes` and 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  $\Delta_x$ , the parallel CTI adjustment  $\Delta_y$ , and the adjusted pulse heights PHAS\_ADJ are initialized such that

$$\Delta_x[j] = 0, \tag{357}$$

$$\Delta_y[j] = 0, \text{ and} \tag{358}$$

$$\text{PHAS\_ADJ}[j] = \text{PHAS}[j] \tag{359}$$

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

$$n = 1. \tag{360}$$

iii. The temporary variables  $\Delta'_x$ ,  $\Delta'_y$ , and PHAS\_ADJ' are set such that

$$\Delta'_x[j] = \Delta_x[j], \quad (361)$$

$$\Delta'_y[j] = \Delta_y[j], \text{ and} \quad (362)$$

$$\text{PHAS\_ADJ}'[j] = \text{PHAS\_ADJ}[j] \quad (363)$$

for each element  $j$ .

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

$$\Delta_x[0] = c_x[0]s_x\rho_x[0]V_x[0], \quad (364)$$

$$\Delta_x[1] = c_x[1]s_x\rho_x[1]V_x[1] - c'_x[0]s_x\rho_x[0]V_x[0], \quad (365)$$

$$\Delta_x[2] = c_x[2]s_x\rho_x[2]V_x[2] - c'_x[1]s_x\rho_x[1]V_x[1], \quad (366)$$

$$\Delta_x[3] = c_x[3]s_x\rho_x[3]V_x[3], \quad (367)$$

$$\Delta_x[4] = c_x[4]s_x\rho_x[4]V_x[4] - c'_x[3]s_x\rho_x[3]V_x[3], \quad (368)$$

$$\Delta_x[5] = c_x[5]s_x\rho_x[5]V_x[5] - c'_x[4]s_x\rho_x[4]V_x[4], \quad (369)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6], \quad (370)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[6]s_x\rho_x[6]V_x[6], \text{ and} \quad (371)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8] - c'_x[7]s_x\rho_x[7]V_x[7], \quad (372)$$

where

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

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

$$s_x = 1 + \text{TCTIX}(T - \text{FP\_TEMPO}), \quad (373)$$

$s_x$  is a temperature dependent scaling factor,  
TCTIX is the `CCD_ID` dependent value in the column TCTIX of the `ctifile`,  
FP\_TEMPO is the name of a keyword in the `ctifile`,

$$T = \left( \frac{t' - t'_k}{t'_{k+1} - t'_k} \right) (\text{FP\_TEMP}_{k+1} - \text{FP\_TEMP}_k) + \text{FP\_TEMP}_k, \quad (374)$$

{  $T$  is the time dependent focal plane temperature,

$$t' = t + \text{TIMEDEL}_{\text{in}} (\text{TIMEPIXR}_{\text{evt}} - 0.5), \quad (375)$$

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

$$t'_k = \text{TIME}_k + \text{TIMEDEL}_{\text{mtl}} (\text{TIMEPIXR}_{\text{mtl}} - 0.5), \quad (376)$$

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

$$t'_{k+1} = \text{TIME}_{k+1} + \text{TIMEDEL}_{\text{mtl}} (\text{TIMEPIXR}_{\text{mtl}} - 0.5), \quad (377)$$

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

$$\rho_x[j] = \text{serial trap density}, \quad (378)$$

{  $\rho_x[j]$  depends upon the CCD\_ID and upon the CHIPX and nint(CHIPY\_ADJ)  
 coordinates associated with element  $j$  of PHAS\_ADJ[j] (see Fig. 1),

$$V_x[j] = \left( \frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME\_X}_{l+1} - \text{VOLUME\_X}_l) + \text{VOLUME\_X}_l, \quad (379)$$

{  $\text{PHA}_l$  is the  $l^{\text{th}}$  element of the column PHA in the `ctifile`,  
 $\text{PHA}_l$  (and  $\text{PHA}_{l+1}$ ) are CCD\_ID dependent,  
 $\text{PHA}_l \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j]$ ,  
 If  $\text{PHA}_l > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j]$  for  $l = 0$ , then  $l = 0$ ,  
 $\text{PHA}_{l+1}$  is the  $(l+1)^{\text{th}}$  element of the column PHA in the `ctifile`,  
 $\text{PHA}_{l+1} > \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j]$ ,  
 If  $\text{PHA}_{l+1} \leq \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j]$  for  $l = n$ , where  $n$  is the last  
 element, then  $l = n$ ,  
 $\text{VOLUME\_X}_l$  is the  $l^{\text{th}}$  element of the column VOLUME\_X in the `ctifile`,  
 $\text{VOLUME\_X}_l$ , which is CCD\_ID dependent, is associated with  $\text{PHA}_l$ ,  
 $\text{VOLUME\_X}_{l+1}$  is the  $(l+1)^{\text{th}}$  element of the column VOLUME\_X in the  
`ctifile`, and  
 $\text{VOLUME\_X}_{l+1}$ , which is CCD\_ID dependent, is associated with  $\text{PHA}_{l+1}$

B. If there is a serial CTI trap-density map in the `ctifile` for CCD\_ID and NODE\_ID = 1 or 3, then the values of  $\Delta_x$  are given by

$$\Delta_x[0] = c_x[0] s_x \rho_x[0] V_x[0] - c'_x[1] s_x \rho_x[1] V_x[1], \quad (380)$$

$$\Delta_x[1] = c_x[1] s_x \rho_x[1] V_x[1] - c'_x[2] s_x \rho_x[2] V_x[2], \quad (381)$$

$$\Delta_x[2] = c_x[2] s_x \rho_x[2] V_x[2], \quad (382)$$

$$\Delta_x[3] = c_x[3] s_x \rho_x[3] V_x[3] - c'_x[4] s_x \rho_x[4] V_x[4], \quad (383)$$

$$\Delta_x[4] = c_x[4] s_x \rho_x[4] V_x[4] - c'_x[5] s_x \rho_x[5] V_x[5], \quad (384)$$

$$\Delta_x[5] = c_x[5] s_x \rho_x[5] V_x[5], \quad (385)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6] - c'_x[7]s_x\rho_x[7]V_x[7], \quad (386)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[8]s_x\rho_x[8]V_x[8], \text{ and} \quad (387)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8], \quad (388)$$

where

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

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

and  $s_x$ ,  $T$ ,  $t'$ ,  $t'_k$ ,  $t'_{k+1}$ ,  $\rho_x[j]$ , and  $V_x[j]$  are given by equations. 373, 374, 375, 376, 377, 378, and 379, respectively.

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

$$\Delta_y[0] = c_y[0]s_y\rho_y[0]V_y[0], \quad (389)$$

$$\Delta_y[1] = c_y[1]s_y\rho_y[1]V_y[1], \quad (390)$$

$$\Delta_y[2] = c_y[2]s_y\rho_y[2]V_y[2], \quad (391)$$

$$\Delta_y[3] = c_y[3]s_y\rho_y[3]V_y[3] - c'_y[0]s_y\rho_y[0]V_y[0], \quad (392)$$

$$\Delta_y[4] = c_y[4]s_y\rho_y[4]V_y[4] - c'_y[1]s_y\rho_y[1]V_y[1], \quad (393)$$

$$\Delta_y[5] = c_y[5]s_y\rho_y[5]V_y[5] - c'_y[2]s_y\rho_y[2]V_y[2], \quad (394)$$

$$\Delta_y[6] = c_y[6]s_y\rho_y[6]V_y[6] - c'_y[3]s_y\rho_y[3]V_y[3], \quad (395)$$

$$\Delta_y[7] = c_y[7]s_y\rho_y[7]V_y[7] - c'_y[4]s_y\rho_y[4]V_y[4], \text{ and} \quad (396)$$

$$\Delta_y[8] = c_y[8]s_y\rho_y[8]V_y[8] - c'_y[5]s_y\rho_y[5]V_y[5], \quad (397)$$

where

$$\begin{aligned}
c_y[j] &= \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j\text{)}, \\ \text{FRCTRLY} \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8\text{)}, \end{array} \right. \\ \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2\text{)} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8\text{)}, \end{array} \right. \end{cases} \\
c'_y[j] &= \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] < \text{split threshold or} \\ j \rightarrow \text{CHIPY} = 1 \text{ or } 1024 \\ \text{(for } j = 1, 2, 3, 4, 5\text{)}, \\ \text{FRCTRLY} \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5\text{)}, \end{array} \right. \\ \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5\text{)}, \end{array} \right. \end{cases} \\
s_y &= 1 + \text{TCTIY}(T - \text{FP\_TEMPO}), \tag{398}
\end{aligned}$$

$$\begin{cases} s_y \text{ is a temperature dependent scaling factor,} \\ \text{TCTIY is the CCD\_ID dependent value in the column TCTIY of the} \\ \text{ctifile,} \\ \text{FP\_TEMPO is the name of a keyword in the ctifile,} \end{cases}$$

$$\rho_y[j] = \text{parallel trap density}, \tag{399}$$

$$\begin{cases} \rho_y[j] \text{ depends upon the CCD\_ID and upon the CHIPX and nint(CHIPY\_ADJ)} \\ \text{coordinates associated with element } j \text{ of PHAS\_ADJ}[j] \text{ (see Fig. 1),} \end{cases}$$

$$V_y[j] = \left( \frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME\_Y}_{l+1} - \text{VOLUME\_Y}_l) + \text{VOLUME\_Y}_l, \tag{400}$$

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

and  $T$ ,  $t'$ ,  $t'_k$ , and  $t'_{k+1}$ , are given by equations. 374, 375, 376, and 377, respectively.  
vi. The CTI-adjusted pulse heights

$$\text{PHAS\_ADJ}[j] = \text{PHAS}[j] + \Delta_x[j] + \Delta_y[j] \quad (401)$$

for all  $j$ .

vii. A. If

$$|\text{PHAS\_ADJ}'[j] - \text{PHAS\_ADJ}[j]| < \text{cticonverge} \text{ (for all } j \text{) and} \quad (402)$$

$$n \leq \text{max\_cti\_iter}, \quad (403)$$

then the computation of PHAS\_ADJ is complete for the event.

B. If

$$|\text{PHAS\_ADJ}'[j] - \text{PHAS\_ADJ}[j]| \geq \text{cticonverge} \text{ (for one or more } j \text{) and} \quad (404)$$

$$n < \text{max\_cti\_iter}, \quad (405)$$

then  $n = n + 1$  and steps 1.5.15(a)iii–1.5.15(a)vii are repeated.

C. If

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

$$n \geq \text{max\_cti\_iter}, \quad (407)$$

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

## 16. FLTGRADE:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ and} \quad (408)$$

$$\text{apply\_cti} = \text{yes}, \quad (409)$$

then

$$\text{FLTGRADE} = c_f[0] + 2c_f[1] + 4c_f[2] + 8c_f[3] + 16c_f[5] + 32c_f[6] + 64c_f[7] + 128c_f[8], \quad (410)$$

where

$$c_f[j] = \begin{cases} 0 & \text{if PHAS\_ADJ}[j] < \text{split threshold} \\ 1 & \text{otherwise,} \end{cases} \quad (411)$$

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

(b) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT} \text{ and} \quad (412)$$

$$\text{apply\_cti} = \text{no}, \quad (413)$$

then

$$\text{FLTGRADE} = c_f[0] + 2c_f[1] + 4c_f[2] + 8c_f[3] + 16c_f[5] + 32c_f[6] + 64c_f[7] + 128c_f[8], \quad (414)$$

where

$$c_f[j] = \begin{cases} 0 & \text{if PHAS}[j] < \text{split threshold} \\ 0 & \text{if PHAS}[j] > 4095 \\ 0 & \text{if PHAS}[j] > \text{PHAS}[4] \text{ for } j = 0-3 \\ 0 & \text{if PHAS}[j] \geq \text{PHAS}[4] \text{ for } j = 5-8 \\ 1 & \text{otherwise.} \end{cases} \quad (415)$$



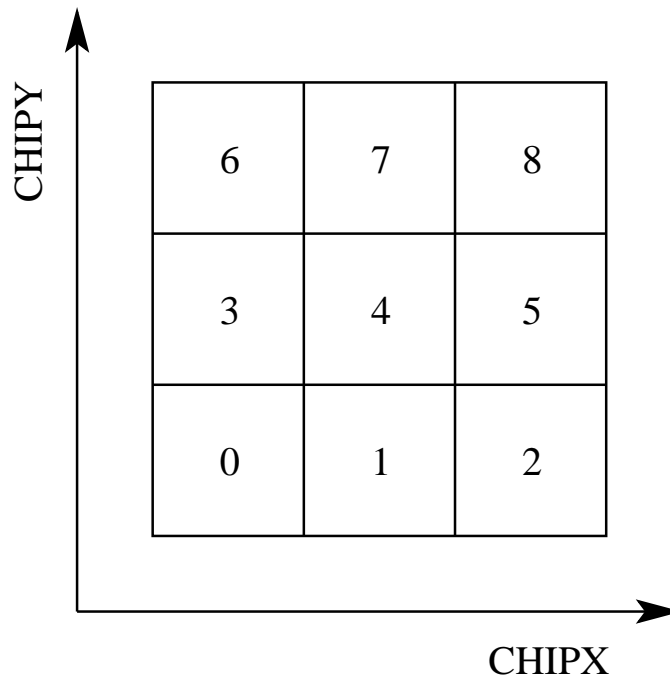


Figure 1: The relative CHIPX and CHIPY coordinates of the nine elements  $j = 0-8$  of a  $3 \text{ pixel} \times 3 \text{ pixel}$  event island PHAS[ $j$ ] or PHAS\_ADJ[ $j$ ].

(c) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (416)$$

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

17. GRADE:

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

- 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

$$\text{FLTGRADE}_{\text{grade},i} = \text{FLTGRADE}, \quad (417)$$

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

iii. The GRADE of the event is given by

$$\text{GRADE} = \text{GRADE}_{\text{grade},i}, \quad (418)$$

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

18. PHA\_RO:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_FAINT}, \quad (419)$$

then

$$\text{PHA\_RO} = \sum_{j=0}^8 \beta[j]p[j], \quad (420)$$

where

i.

$$p[j] = \text{PHAS}[j]. \quad (421)$$

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

iii.

$$\beta[j] = 0 \quad \text{if} \quad p[j] < \text{split threshold}. \quad (422)$$

iv.

$$\beta[j] = 0 \quad \text{if} \quad \begin{cases} p[j] > p[4] & (\text{for } j = 0-3) \\ p[j] \geq p[4] & (\text{for } j = 5-8) \end{cases} \quad (423)$$

v. If CORNERS = -1, then

$$\beta[0] = \beta[2] = \beta[6] = \beta[8] = 0. \quad (424)$$

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

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[3] = 0. \quad (425)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{and} \quad \beta[5] = 0. \quad (426)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (427)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{and} \quad \beta[7] = 0. \quad (428)$$

viii. If CORNERS = 2, then

$$\beta[0] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[3] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (429)$$

$$\beta[2] = 0 \quad \text{if} \quad \beta[1] = 0 \quad \text{or} \quad \beta[5] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (430)$$

$$\beta[6] = 0 \quad \text{if} \quad \beta[3] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (431)$$

$$\beta[8] = 0 \quad \text{if} \quad \beta[5] = 0 \quad \text{or} \quad \beta[7] = 0 \quad \text{or} \quad \text{GRADE} \neq 6. \quad (432)$$

(b) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (433)$$

then

i. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (434)$$

then the value of PHA\_RO for the event is the value of PHA in the infile.

ii. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1} \quad \text{or} \quad (435)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad \text{or} \quad (436)$$

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

then the value of PHA\_RO for the event is the value of PHA\_RO in the infile.

19. 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.

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

iv. If the CTI adjustment is not performed, then

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

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

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

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

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

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

viii. If CORNERS = 2, then

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

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

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

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

(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

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

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

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

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

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

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

$$\left\{ \begin{array}{l} r \text{ is the row of the } \text{tgainfile} \text{ where} \\ \left\{ \begin{array}{l} \text{CCD\_ID}[r] = \text{CCD\_ID}, \\ \text{CHIPX\_LO}[r] \leq \text{CHIPX}, \\ \text{CHIPX\_HI}[r] \geq \text{CHIPX}, \\ \text{CHIPY\_LO}[r] \leq \text{nint}(\text{CHIPY\_ADJ}), \text{ and} \\ \text{CHIPY\_HI}[r] \geq \text{nint}(\text{CHIPY\_ADJ}). \end{array} \right. \\ m \text{ is the element of row } r \text{ where} \\ \left\{ \begin{array}{l} \text{PHA}_m[r] \leq \text{PHA} \text{ and} \\ \text{PHA}_{m+1}[r] > \text{PHA}. \\ \text{If } \text{PHA} < \text{PHA}_m[r] \text{ for } m = 0, \text{ then } m = 0. \\ \text{If } \text{PHA} \geq \text{PHA}_m[r] \text{ for } m = M \text{ and } M \text{ is the last element of } \text{PHA}[r], \\ \text{then } m = M - 1. \end{array} \right. \end{array} \right. \quad (461)$$

The `tgainfile` includes a binary table with columns named  
`CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, `CHIPY_HI`, `PHA`, `DELTPHA1`, and  
`DELTPHA2`.

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

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

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

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

(d) If

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

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

20. CORN\_PHA:

(a) If

$$\text{DATAMODE}_{\text{in}} = \text{CC33\_GRADED}, \quad (467)$$

then the value of CORN\_PHA is read from the `infile`.

21. ENERGY:

(a) If the parameter `calculate_pi` = yes and the parameter `gainfile` is specified and  $\text{PHA} > 0$ , 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  $\Delta p$  is computed over the interval from  $[-0.5, +0.5)$ .
- iii. The element  $j$  of row  $i$  of  $\text{PHA}_{\text{gain}}$  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  $\text{PHA}_{\text{gain}}$  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 and the parameter `gainfile` is specified and  $\text{PHA} \leq 0$ , 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  $\text{ENERGY} = 0$ .

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

## 23. `pix_adj`:

- (a) centroid:

i. If

$$\text{pix\_adj} = \text{centroid} \quad (478)$$

and

$$\text{DATAMODE} = \text{CC33\_FAINT} \text{ or} \quad (479)$$

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

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

$$\text{DATAMODE} = \text{VFAINT}, \quad (482)$$

then

$$\text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} - w'[0] + w'[2] - w'[3] + w'[5] - w'[6] + w'[8] \text{ and} \quad (483)$$

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

where

$$w'[j] = \frac{w[j]}{\sum_{j=0}^8 w[j]}, \quad (485)$$

$$w[j] = \begin{cases} p[j] & \text{if the pixel is valid} \\ 0 & \text{if the pixel is invalid,} \end{cases} \quad (486)$$

$$p[j] = \begin{cases} \text{PHAS\_ADJ}[j] & \text{if apply\_cti} = \text{yes} \\ \text{PHAS}[j] & \text{if apply\_cti} = \text{no,} \end{cases} \quad (487)$$

and the pixel is invalid if

$$\beta[j] = 0 \text{ or} \quad (488)$$

$$\text{STATUS}[0] = 1 \text{ or} \quad (489)$$

$$\text{STATUS}[1] = 1 \text{ or} \quad (490)$$

$$\text{STATUS}[2] = 1 \text{ or} \quad (491)$$

$$\text{STATUS}[3] = 1 \text{ or} \quad (492)$$

$$\text{STATUS}[4] = 1 \text{ or} \quad (493)$$

$$\text{STATUS}[11] = 1 \text{ or} \quad (494)$$

$$\text{STATUS}[13] = 1 \text{ or} \quad (495)$$

$$\text{STATUS}[14] = 1 \text{ or} \quad (496)$$

$$\text{STATUS}[15] = 1 \text{ or} \quad (497)$$

$$\text{STATUS}[16] = 1. \quad (498)$$

ii. If

$$\text{pix\_adj} = \text{centroid} \text{ and} \quad (499)$$

$$\text{DATAMODE} = \text{CC33\_FAINT}, \quad (500)$$

then

$$\text{TIME} = \text{TIME} + (w'[0] + w'[1] + w'[2] - w'[6] - w'[7] - w'[8]) \times \text{TIMEDEL}_{\text{in}}. \quad (501)$$

Note that it is possible for the centroid algorithm to yield adjustments to CHIPX\_ADJ and/or CHIPY\_ADJ that are greater than half a pixel. However, the adjustment cannot equal or exceed one pixel.

(b) edser:

i. If

$$\text{pix\_adj} = \text{edser} \quad (502)$$

and

$$\text{DATAMODE} = \text{CC33\_FAINT} \text{ or} \quad (503)$$

$$\text{DATAMODE} = \text{CC33\_GRADED} \text{ or} \quad (504)$$

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

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

$$\text{DATAMODE} = \text{GRADED} \text{ or} \quad (507)$$

$$\text{DATAMODE} = \text{VFAINT}, \quad (508)$$

then

$$\text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} + \left( \frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta X[k+1] - \Delta X[k]) + \Delta X[k] \quad (509)$$

$$\text{CHIPY\_ADJ} = \text{CHIPY\_ADJ} + \left( \frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k], \quad (510)$$

where  $E[k]$  and  $E[k+1]$ ,  $\Delta X[k]$  and  $\Delta X[k+1]$ , and  $\Delta Y[k]$  and  $\Delta Y[k+1]$  are the  $k$  and  $(k+1)^{th}$  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

$$\text{ENERGY} \geq E[k] \text{ and} \quad (511)$$

$$\text{ENERGY} < E[k+1]. \quad (512)$$

Note that if

$$\text{ENERGY} \leq E[0], \quad (513)$$

then  $k = 0$ . Similarly, if

$$\text{ENERGY} \geq E[\text{NPOINTS}_{\text{subpix}} - 2], \quad (514)$$

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

ii. If

$$\text{pix\_adj} = \text{edser} \quad (515)$$

and

$$\text{DATAMODE} = \text{CC33\_FAINT} \text{ or} \quad (516)$$

$$\text{DATAMODE} = \text{CC33\_GRADED}, \quad (517)$$

then

$$\text{TIME} = \text{TIME} - \left( \left( \frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k] \right) \times \text{TIMEDEL}_i \quad (518)$$

(c) none:

If

$$\text{pix\_adj} = \text{none}, \quad (519)$$

then

$$\text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} \text{ and} \quad (520)$$

$$\text{CHIPY\_ADJ} = \text{CHIPY\_ADJ} \text{ and} \quad (521)$$

$$\text{TIME} = \text{TIME}. \quad (522)$$

No sub-pixel adjustments are applied to the values of `CHIPX_ADJ` and `CHIPY_ADJ` (for timed exposure mode) or `CHIPX_ADJ` and `TIME` (for continuous-clocking mode).

(d) randomize:

i. If

$$\text{pix\_adj} = \text{randomize}, \quad (523)$$

then

$$\text{CHIPX\_ADJ} = \text{CHIPX\_ADJ} + \epsilon_x \text{ and} \quad (524)$$

$$\text{CHIPY\_ADJ} = \text{CHIPY\_ADJ} + \epsilon_y, \quad (525)$$

where  $\epsilon_x$  and  $\epsilon_y$  are a uniform random deviates in the range  $[-0.5, +0.5)$  pixel.

ii. If

$$\text{DATAMODE} = \text{CC33\_FAINT} \text{ or} \quad (526)$$

$$\text{DATAMODE} = \text{CC33\_GRADED}, \quad (527)$$

then

$$\text{TIME} = \text{TIME} - \epsilon_y \times \text{TIMEDEL}_{\text{in}}. \quad (528)$$

(e) If

$$\text{CHIPX\_ADJ} < 0.5, \quad (529)$$

then

$$\text{CHIPX\_ADJ} = 1. \quad (530)$$

(f) If

$$\text{CHIPX\_ADJ} \geq 1024.5, \quad (531)$$

then

$$\text{CHIPX\_ADJ} = 1024. \quad (532)$$

(g) If

$$\text{CHIPY\_ADJ} < 0.5, \quad (533)$$

then

$$\text{CHIPY\_ADJ} = 1. \quad (534)$$

(h) If

$$\text{CHIPY\_ADJ} \geq 1024.5, \quad (535)$$

then

$$\text{CHIPY\_ADJ} = 1024. \quad (536)$$



24. X and Y:

(a) If

$$\text{stop} = \text{sky}, \quad (537)$$

then

i. If

$$\text{DATAMODE} = \text{FAINT or} \quad (538)$$

$$\text{DATAMODE} = \text{FAINT\_BIAS or} \quad (539)$$

$$\text{DATAMODE} = \text{GRADED or} \quad (540)$$

$$\text{DATAMODE} = \text{VFAINT}, \quad (541)$$

then the values of X and Y are computed using the real-valued coordinates CHIPX\_ADJ and CHIPY\_ADJ and the orientation of the telescope (i.e. RA, DEC, and ROLL) at the time TIME.

ii. If

$$\text{DATAMODE} = \text{CC33\_FAINT or} \quad (542)$$

$$\text{DATAMODE} = \text{CC33\_GRADED}, \quad (543)$$

then the values of X and Y are computed using the real-valued coordinates CHIPX\_ADJ and CHIPY\_TARG and the orientation of the telescope (i.e. RA, DEC, and ROLL) at the time TIME.

iii. If

$$\text{acaofffile} \neq \text{none and} \quad (544)$$

$$\text{acaofffile} \neq \text{NONE} \quad (545)$$

and

$$\text{TIME} < \text{TIME}_{\min} \text{ or} \quad (546)$$

$$\text{TIME} \geq \text{TIME}_{\max}, \quad (547)$$

then

$$X = \text{NaN and} \quad (548)$$

$$Y = \text{NaN}. \quad (549)$$

iv. If

$$\text{fltfile} \neq \text{none and} \quad (550)$$

$$\text{fltfile} \neq \text{NONE} \quad (551)$$

and TIME is not in a good-time interval, then

$$X = \text{NaN and} \quad (552)$$

$$Y = \text{NaN}. \quad (553)$$

25. SKY\_1D:

(a) If

$$\text{DATAMODE} = \text{CC33\_FAINT or} \quad (554)$$

$$\text{DATAMODE} = \text{CC33\_GRADED} \quad (555)$$

and

$$\text{stop} = \text{sky}, \quad (556)$$

then

i. The value of `SKY_1D` is computed using the real-valued coordinates `CHIPX_ADJ` and `CHIPY_TARG` and the orientation of the telescope (i.e. `RA`, `DEC`, and `ROLL`) at the time `TIME`.

ii. If

$$\text{acaofffile} \neq \text{none and} \quad (557)$$

$$\text{acaofffile} \neq \text{NONE} \quad (558)$$

and

$$\text{TIME} < \text{TIME}_{\min} \text{ or} \quad (559)$$

$$\text{TIME} \geq \text{TIME}_{\max}, \quad (560)$$

then

$$\text{SKY\_1D} = \text{NaN}. \quad (561)$$

iii. If

$$\text{fltfile} \neq \text{none and} \quad (562)$$

$$\text{fltfile} \neq \text{NONE} \quad (563)$$

and `TIME` is not in a good-time interval, then

$$\text{SKY\_1D} = \text{NaN}. \quad (564)$$

## 26. `DETX` and `DETY`:

(a) If

$$\text{stop} = \text{det or} \quad (565)$$

$$\text{stop} = \text{tan or} \quad (566)$$

$$\text{stop} = \text{sky}, \quad (567)$$

then

i. The values of `DETX` and `DETY` are computed using the real-valued coordinates `CHIPX_ADJ` and `CHIPY_ADJ` and the orientation of the SIM (i.e. `DY`, `DZ`, and `DTHETA`) at the time `TIME`.

ii. If

$$\text{acaofffile} \neq \text{none and} \quad (568)$$

$$\text{acaofffile} \neq \text{NONE} \quad (569)$$

and

$$\text{TIME} < \text{TIME}_{\min} \text{ or} \quad (570)$$

$$\text{TIME} \geq \text{TIME}_{\max}, \quad (571)$$

then

$$\text{DETX} = \text{NaN and} \quad (572)$$

$$\text{DETY} = \text{NaN}. \quad (573)$$

iii. If

$$\text{fltfile} \neq \text{none and} \quad (574)$$

$$\text{fltfile} \neq \text{NONE} \quad (575)$$

and TIME is not in a good-time interval, then

$$\text{DETX} = \text{NaN and} \quad (576)$$

$$\text{DETY} = \text{NaN.} \quad (577)$$

27. TDETX and TDETY:

(a) If

$$\text{stop} = \text{tdet or} \quad (578)$$

$$\text{stop} = \text{det or} \quad (579)$$

$$\text{stop} = \text{tan or} \quad (580)$$

$$\text{stop} = \text{sky} \quad (581)$$

then

i. The values of TDETX and TDETY are computed using the values of `nint(CHIPX_ADJ)` and `nint(CHIPY_ADJ)`. Here, “nint” indicates that the real-valued coordinate is rounded to the nearest integer.

ii. If

$$\text{acaofffile} \neq \text{none and} \quad (582)$$

$$\text{acaofffile} \neq \text{NONE} \quad (583)$$

and

$$\text{TIME} < \text{TIME}_{\min} \text{ or} \quad (584)$$

$$\text{TIME} \geq \text{TIME}_{\max}, \quad (585)$$

then

$$\text{TDETX} = \text{NULL and} \quad (586)$$

$$\text{TDETY} = \text{NULL.} \quad (587)$$

iii. If

$$\text{fltfile} \neq \text{none and} \quad (588)$$

$$\text{fltfile} \neq \text{NONE} \quad (589)$$

and TIME is not in a good-time interval, then

$$\text{TDETX} = \text{NULL and} \quad (590)$$

$$\text{TDETY} = \text{NULL.} \quad (591)$$

#### 1.5.4 Write outfile

1. PIX\_ADJ:

(a) If  $\text{pix\_adj} = \text{centroid}$ , (592)

then

$\text{PIX\_ADJ} = \text{CENTROID}$ . (593)

(b) If

$\text{pix\_adj} = \text{edser}$ , (594)

then

$\text{PIX\_ADJ} = \text{EDSER}$ . (595)

(c) If

$\text{pix\_adj} = \text{none}$ , (596)

then

$\text{PIX\_ADJ} = \text{NONE}$ . (597)

(d) If

$\text{pix\_adj} = \text{randomize}$ , (598)

then

$\text{PIX\_ADJ} = \text{RANDOMIZE}$ . (599)

## 2. RAND\_SKY:

(a) If

$\text{pix\_adj} = \text{centroid}$ , (600)

then

$\text{RAND\_SKY} = 0.0$ . (601)

(b) If

$\text{pix\_adj} = \text{edser}$ , (602)

then

$\text{RAND\_SKY} = 0.0$ . (603)

(c) If

$\text{pix\_adj} = \text{none}$ , (604)

then

$\text{RAND\_SKY} = 0.0$ . (605)

(d) If

`pix_adj = randomize,` (606)

then

`RAND_SKY = 0.5.` (607)

3. `TIME_ADJ`:

(a) Timed-exposure mode:

If

`DATAMODEin = FAINT or` (608)

`DATAMODEin = FAINT_BIAS or` (609)

`DATAMODEin = GRADED or` (610)

`DATAMODEin = VFAINT,` (611)

then

`TIME_ADJ = NONE.` (612)

(b) Continuous-clocking mode:

i. Set

`TIME_ADJ = MIDCHIP.` (613)

ii. If

`OBS_MODE = pointing or` (614)

`OBS_MODE = POINTING` (615)

and

`acaofffile ≠ none and` (616)

`acaofffile ≠ NONE,` (617)

then

`TIME_ADJ = TARGET.` (618)

iii. If

`OBS_MODE = pointing or` (619)

`OBS_MODE = POINTING` (620)

and

`CONTENTin = TGEVT1,` (621)

then

`TIME_ADJ = TARGET.` (622)

## 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)?
- Make sure that the STATUS bits are unset and reset properly.
- What about aoff and soff files instead of asol files?
- Are the  $\beta$  in PHA\_RO the same as the  $\beta$  in PHA?
- Should something be done about SKY\_1D?