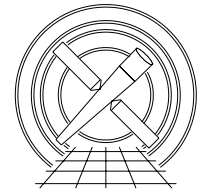




MIT Kavli Institute



Chandra X-Ray Center

MEMORANDUM

June 15, 2016

To: Jonathan McDowell, SDS Group Leader
From: Glenn E. Allen, SDS
Subject: `acis_process_events` spec
Revision: 4.13
URL: <http://space.mit.edu/CXC/docs/docs.html#ape>
File: `/nfs/inconceivable/d0/sds/specs/acis_process_events/ape_spec.4.13.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. If

$$\text{obsfile} = \text{NONE}, \tag{1}$$

then `obsfile` is changed to “none”.

ii. Existence:

If

$$\text{obsfile} \neq \text{none} \tag{2}$$

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

iii. Permission:

If

$$\text{obsfile} \neq \text{none} \quad (3)$$

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.

iv. OBS_MODE:

If

$$\text{obsfile} \neq \text{none} \quad (4)$$

then

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

B. If the `obsfile` includes the keyword `OBS_MODE` then `OBS_MODEin` is set to `OBS_MODE`.

C. If

$$\text{OBS_MODE}_{in} = \text{POINTING}, \quad (5)$$

then `OBS_MODEin` is set to “pointing”.

D. If

$$\text{OBS_MODE}_{in} = \text{SECONDARY}, \quad (6)$$

then `OBS_MODEin` is set to “secondary”.

E. If

$$\text{OBS_MODE}_{in} \neq \text{none and} \quad (7)$$

$$\text{OBS_MODE}_{in} \neq \text{pointing and} \quad (8)$$

$$\text{OBS_MODE}_{in} \neq \text{secondary}, \quad (9)$$

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

2. `acaofffile`:

(a) Validation:

If

$$\text{OBS_MODE}_{in} = \text{pointing}, \quad (10)$$

then

i. If

$$\text{acaofffile} = \text{NONE}, \quad (11)$$

then `acaofffile` is set to “none.”

ii. Setting:

If

$$\text{acaofffile} = \text{none}, \quad (12)$$

then `acis_process_events` produces a warning message.

iii. Existence:

If

$$\text{acaofffile} \neq \text{none} \quad (13)$$

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

iv. 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.”

v. CONTENT:

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

$$\text{CONTENT} = \text{ASPSOL} \text{ or} \quad (14)$$

$$\text{CONTENT} = \text{OBCSOL}, \quad (15)$$

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

vi. Keyword:

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

vii. Columns:

If HDU h_{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 $\text{ROLL}_{\text{acaoff}}$.

viii. 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

$$\text{OBS_MODE}_{\text{in}} = \text{pointing}, \quad (16)$$

then

i. If

$$\text{alignmentfile} = \text{NONE}, \quad (17)$$

then `alignmentfile` is changed to “none.”

ii. Setting:

If

$$\text{alignmentfile} = \text{none}, \quad (18)$$

then `acis_process_events` produces a warning message.

iii. Existence:

If

$$\text{alignmentfile} \neq \text{none} \quad (19)$$

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

iv. 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.”.

v. CONTENT:

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

$$\text{CONTENT} = \text{ASPSOL or} \quad (20)$$

$$\text{CONTENT} = \text{OBCSOL,} \quad (21)$$

then `acis-process_events` produces a warning message and `alignmentfile` is set to “none.”.

vi. 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.”.

vii. 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.”.

viii. 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. 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 $\text{OBS_MODE}_{\text{in}} = \text{none}$ and HDU h_{in} of the `infile` includes the keyword OBS_MODE, then

A. $\text{OBS_MODE}_{\text{in}}$ is set to OBS_MODE.

B. If

$$\text{OBS_MODE}_{\text{in}} = \text{POINTING,} \quad (22)$$

then $\text{OBS_MODE}_{\text{in}}$ is set to “pointing”.

C. If

$$\text{OBS_MODE}_{\text{in}} = \text{SECONDARY,} \quad (23)$$

then $\text{OBS_MODE}_{\text{in}}$ is set to “secondary”.

D. If

$$\text{OBS_MODE}_{\text{in}} \neq \text{none and} \quad (24)$$

$$\text{OBS_MODE}_{\text{in}} \neq \text{pointing and} \quad (25)$$

$$\text{OBS_MODE}_{\text{in}} \neq \text{secondary,} \quad (26)$$

then $\text{OBS_MODE}_{\text{in}}$ 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

$$\text{DATAMODE} \neq \text{CC33_FAINT} \text{ and} \quad (27)$$

$$\text{DATAMODE} \neq \text{CC33_GRADED} \text{ and} \quad (28)$$

$$\text{DATAMODE} \neq \text{FAINT} \text{ and} \quad (29)$$

$$\text{DATAMODE} \neq \text{FAINT_BIAS} \text{ and} \quad (30)$$

$$\text{DATAMODE} \neq \text{GRADED} \text{ and} \quad (31)$$

$$\text{DATAMODE} \neq \text{VFAINT}, \quad (32)$$

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

$$\text{CONTENT} = \text{EVT0} \text{ or} \quad (33)$$

$$\text{CONTENT} = \text{EVT1} \text{ or} \quad (34)$$

$$\text{CONTENT} = \text{TGEVT1} \text{ or} \quad (35)$$

$$\text{CONTENT} = \text{EVT2}, \quad (36)$$

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

iv. TIME:

If

$$\text{DATAMODE}_{in} = \text{CC33_FAINT} \text{ or} \quad (37)$$

$$\text{DATAMODE}_{in} = \text{CC33_GRADED} \quad (38)$$

and 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

$$\text{DATAMODE}_{in} = \text{CC33_FAINT} \text{ or} \quad (39)$$

$$\text{DATAMODE}_{in} = \text{CC33_GRADED} \quad (40)$$

and

$$\text{CONTENT}_{in} = \text{EVT1} \text{ or} \quad (41)$$

$$\text{CONTENT}_{in} = \text{TGEVT1} \text{ or} \quad (42)$$

$$\text{CONTENT}_{in} = \text{EVT2} \quad (43)$$

and 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. CCD_ID:

A. If

$$\text{CONTENT}_{in} = \text{EVT0} \quad (44)$$

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

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (45)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (46)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (47)$$

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`.

vii. CCDX:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (48)$$

and HDU h_{in} of the `infile` does not include the column `CCDX` and does not include the column `CHIPX`, then `acis_process_events` exits with an error message. Hereafter, these columns are referred to as `CCDXin` and `CHIPXin`, respectively.

viii. CHIPX:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (49)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (50)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (51)$$

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`.

ix. CCDY:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (52)$$

and

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

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

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (55)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \quad (56)$$

and HDU h_{in} of the `infile` does not include the column `CCDY` and does not include the column `CHIPY`, then `acis_process_events` exits with an error message. Hereafter, these columns are referred to as `CCDYin` and `CHIPYin`, respectively.

x. TROW:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT0} \quad (57)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (58)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (59)$$

and HDU h_{in} of the `infile` does not include the column `TROW` and does not include the column `CHIPY`, then `acis_process_events` exits with an error message. Hereafter, these columns are referred to as `TROWin` and `CHIPYin`, respectively.

xi. CHIPY:

A. If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (60)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (61)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (62)$$

and 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`.

xii. TIMEDEL:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (63)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (64)$$

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`.

xiii. RA_TARG, DEC_TARG, RA_NOM, DEC_NOM, RA_PNT, DEC_PNT, CHIPY_TG, and TG_M:

If

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \quad (65)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (66)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (67)$$

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 and TG_M:

If

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

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

xiv. TGAINCOR:

If HDU h_{in} of the `infile` does not include the keyword `TGAINCOR`, then this keyword is set to zero (i.e. `FALSE`). Hereafter, this keyword is referred to as `TGAINCORin`.

5. stop:

(a) Lowercase:

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

(b) Validation:

i. Setting:

If

`stop` \neq none and (69)

`stop` \neq chip and (70)

`stop` \neq tdet and (71)

`stop` \neq det and (72)

`stop` \neq tan and (73)

`stop` \neq sky, (74)

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

ii. `OBS_MODE`:

If

`OBS_MODEin` \neq pointing and (75)

`stop` \neq none and (76)

`stop` \neq chip and (77)

`stop` \neq tdet, (78)

then `stop` is changed to “tdet” and `acis_process_events` produces a warning message.

iii. `acaofffile`:

If

`OBS_MODEin` = pointing and (79)

`acaofffile` = none and (80)

`stop` \neq none and (81)

`stop` \neq chip and (82)

`stop` \neq tdet, (83)

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

iv. `alignmentfile`:

If

`OBS_MODEin` = pointing and (84)

`alignmentfile` = none and (85)

`stop` \neq none and (86)

`stop` \neq chip and (87)

`stop` \neq tdet, (88)

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

6. `doevtgrade`:

(a) Lowercase:

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

(b) Validation:

If

`doevtgrade` \neq yes and (89)

`doevtgrade` \neq no, (90)

then `acis_process_events` exits with an error message.

7. `apply_cti`:

(a) Lowercase:

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

(b) Validation:

i. Setting:

If

$$\text{apply_cti} \neq \text{yes and} \tag{91}$$

$$\text{apply_cti} \neq \text{no}, \tag{92}$$

then `acis_process_events` exits with an error message.

ii. PHAS:

If

$$\text{apply_cti} = \text{yes} \tag{93}$$

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

$$\text{apply_cti} = \text{yes and} \tag{94}$$

$$\text{doevtgrade} = \text{no}, \tag{95}$$

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

8. `gradefile`:

(a) Validation:

i. If

$$\text{gradefile} = \text{NONE}, \tag{96}$$

then `gradefile` is changed to “none.”

ii. Existence:

If

$$\text{gradefile} \neq \text{none} \tag{97}$$

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

iii. Permission:

If

$$\text{gradefile} \neq \text{none} \tag{98}$$

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

iv. HDU:

If the `gradefile` does not have an HDU that includes the keyword `CBD10001` and where the keyword includes `DATAMODEin`, then `gradefile` is changed to “none” and `acis_process_events` produces a warning message. Hereafter this HDU is referred to as h_{grade} .

- v. Columns:
If

$$\text{gradefile} \neq \text{none} \tag{99}$$

and HDU h_{grade} of the `gradefile` does not include the columns `FLTGRADE` and `GRADE`, then `gradefile` is changed to “none” and `acis_process_events` produces a warning message. Hereafter these columns are referred to as `FLTGRADEgrade` and `GRADEgrade`, respectively.

9. `badpixfile`:

- (a) Validation:

- i. If

$$\text{badpixfile} = \text{NONE}, \tag{100}$$

then `badpixfile` is changed to “none.”

- ii. Existence:
If

$$\text{badpixfile} \neq \text{none} \tag{101}$$

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

- iii. Permission:
If

$$\text{badpixfile} \neq \text{none} \tag{102}$$

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.

- iv. `CONTENT`:
If

$$\text{badpixfile} \neq \text{none} \tag{103}$$

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

$$\text{CONTENT} = \text{BADPIX} \text{ or} \tag{104}$$

$$\text{CONTENT} = \text{CDB_ACIS_BADPIX}, \tag{105}$$

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

- v. Keyword:
If

$$\text{badpixfile} \neq \text{none} \tag{106}$$

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`.

vi. Columns:

If

$$\text{badpixfile} \neq \text{none} \tag{107}$$

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.

10. `ctifile`:

(a) Validation:

If

$$\text{ctifile} \neq \text{caldb} \text{ and} \tag{108}$$

$$\text{ctifile} \neq \text{CALDB}, \tag{109}$$

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_{cti} with the keyword

$$\text{CONTENT} = \text{CDB_ACIS_CTI}, \tag{110}$$

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

$$\text{clobber} \neq \text{yes} \text{ and} \tag{111}$$

$$\text{clobber} \neq \text{no}, \tag{112}$$

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

ii. Permission:

If

$$\text{clobber} = \text{yes} \tag{113}$$

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

$$\text{clobber} = \text{no} \tag{114}$$

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

$$\text{pix_adj} \neq \text{centroid and} \tag{115}$$
$$\text{pix_adj} \neq \text{edser and} \tag{116}$$
$$\text{pix_adj} \neq \text{none and} \tag{117}$$
$$\text{pix_adj} \neq \text{randomize,} \tag{118}$$

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

ii. `OBS_MODE`:

If

$$\text{OBS_MODE}_{\text{in}} \neq \text{pointing and} \tag{119}$$
$$\text{pix_adj} \neq \text{none,} \tag{120}$$

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

iii. `acaofffile`:

If

$$\text{pix_adj} \neq \text{none} \tag{121}$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \tag{122}$$
$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \tag{123}$$

and

$$\text{acaofffile} = \text{none,} \tag{124}$$

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

iv. `alignmentfile`:

If

$$\text{pix_adj} \neq \text{none} \tag{125}$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \tag{126}$$
$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \tag{127}$$

and

$$\text{alignmentfile} = \text{none,} \tag{128}$$

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

v. stop:

If

$$\text{pix_adj} = \text{centroid or} \quad (129)$$

$$\text{pix_adj} = \text{edser or} \quad (130)$$

$$\text{pix_adj} = \text{randomize} \quad (131)$$

and

$$\text{stop} \neq \text{sky}, \quad (132)$$

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

vi. PHAS:

If

$$\text{pix_adj} = \text{centroid} \quad (133)$$

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

vii. FLTGRADE:

If

$$\text{pix_adj} = \text{edser} \quad (134)$$

and

$$\text{DATAMODE} = \text{CC33_GRADED or} \quad (135)$$

$$\text{DATAMODE} = \text{GRADED} \quad (136)$$

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

$$\text{pix_adj} = \text{edser}, \quad (137)$$

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_{subpix} with the keyword

$$\text{CONTENT} = \text{AXAF_SUBPIX}, \quad (138)$$

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

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \quad (139)$$

and

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

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

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. `TIMEmin` and `TIMEmax`:

If

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \text{ and} \quad (142)$$

$$\text{acaofffile} \neq \text{none} \text{ and} \quad (143)$$

$$\text{alignmentfile} \neq \text{none}, \quad (144)$$

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

$$\text{TIME}_{\text{min}} = \min(\text{TIME}_{\text{acaoff}}) \text{ and} \quad (145)$$

$$\text{TIME}_{\text{max}} = \max(\text{TIME}_{\text{acaoff}}). \quad (146)$$

3. `CHIPX_TARGacaoff,med`, `CHIPY_TARGacaoff,med`:

If

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \quad (147)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (148)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (149)$$

and

$$\text{acaofffile} \neq \text{none} \text{ and} \quad (150)$$

$$\text{alignmentfile} \neq \text{none}, \quad (151)$$

*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.

then the values of $\text{CHIPX_TARG}_{\text{acaoff}}$ and $\text{CHIPY_TARG}_{\text{acaoff}}$ 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 $\text{CHIPX_TARG}_{\text{acaoff}}$ and $\text{CHIPY_TARG}_{\text{acaoff}}$ are processed to obtain the median values:

$$\text{CHIPX_TARG}_{\text{acaoff,med}} = \text{median}(\text{CHIPX_TARG}_{\text{acaoff}}) \text{ and} \quad (152)$$

$$\text{CHIPY_TARG}_{\text{acaoff,med}} = \text{median}(\text{CHIPY_TARG}_{\text{acaoff}}). \quad (153)$$

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

$$\text{doevtgrade} = \text{yes}, \quad (154)$$

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:

- i. If HDU h_{in} of the `infile` includes the `EXPNO`, hereafter referred to as `EXPNOin`, then

$$\text{EXPNO} = \text{EXPNO}_{\text{in}}. \quad (155)$$

- ii. If HDU h_{in} of the `infile` does not include the `EXPNO`, then

$$\text{EXPNO} = \text{NULL}. \quad (156)$$

(b) Validation:

If

$$\text{EXPNO} \neq \text{NULL} \quad (157)$$

and

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

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

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which equation 157 and either equation 158 or 159 is true. These conditions should not occur.

3. CCD_ID:

- (a) Read:
The value of `CCD_ID` for an event is given by `CCD_IDin`.
- (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. If the `infile` includes the column `CHIPX`, then the value of `CHIPX` for an event is given by

$$\text{CHIPX} = \text{CHIPX}_{\text{in}}. \tag{162}$$

- ii. If the `infile` does not include the column `CHIPX`, then the value of `CHIPX` for an event is given by

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

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

$$\text{CHIPX} < 1, \tag{164}$$

then

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

If

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

then

$$\text{CHIPX} = 1024. \tag{167}$$

- ii. Unexpected:
If

$$\text{CHIPX} = 1 \text{ or} \tag{168}$$

$$\text{CHIPX} = 1024, \tag{169}$$

then `acis_process_events` sets `STATUS` bit 0 (of 0–31) to one and increments a counter for the number of unexpected `CHIPX` values. These values should not occur.

5. CHIPX_ADJ:

- (a) Initialize:

$$\text{CHIPX_ADJ} = \text{CHIPX}. \tag{170}$$

6. 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.

7. CHIPY:

(a) Read:

i. If the `infile` includes the column `CHIPY`, then the value of `CHIPY` for an event is given by

$$\text{CHIPY} = \text{CHIPY}_{\text{in}}. \quad (172)$$

ii. If the `infile` does not include the column `CHIPY` and

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

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

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

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

then the value of `CHIPY` for an event is given by

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

iii. If the `infile` does not include the column `CHIPY` and

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

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

then the value of `CHIPY` for an event is given by

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

(b) Validation:

i. Unphysical:

A. Minimum:

If

$$\text{CHIPY} < 1, \quad (181)$$

then

$$\text{CHIPY} = 1. \quad (182)$$

B. Timed exposure mode maximum:

If

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

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

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

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

and

$$\text{CHIPY} > 1024, \quad (187)$$

then

$$\text{CHIPY} = 1024. \quad (188)$$

C. Continuous-clocking mode maximum:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (189)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (190)$$

and

$$\text{CHIPY} > 512, \quad (191)$$

then

$$\text{CHIPY} = 512. \quad (192)$$

ii. Unexpected:

A. FAINT, FAINT_BIAS, or GRADED:

If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT} \text{ or} \quad (193)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS} \text{ or} \quad (194)$$

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

and

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

$$\text{CHIPY} = 1024, \quad (197)$$

then `acis_process_events` sets STATUS bit 0 (of 0–31) to one and increments a counter for the number of unexpected CHIPY values. These values should not occur.

B. VFAINT:

If

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT} \quad (198)$$

and

$$\text{CHIPY} = 1 \text{ or} \quad (199)$$

$$\text{CHIPY} = 2 \text{ or} \quad (200)$$

$$\text{CHIPY} = 1023 \text{ or} \quad (201)$$

$$\text{CHIPY} = 1024, \quad (202)$$

then `acis_process_events` sets STATUS bit 0 (of 0–31) to one and increments a counter for the number of unexpected CHIPY values. These values should not occur.

C. CC33_FAINT or CC33_GRADED:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (203)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (204)$$

and

$$\text{CHIPY} = 1 \text{ or} \quad (205)$$

$$\text{CHIPY} = 512, \quad (206)$$

then `acis_process_events` sets STATUS bit 0 (of 0–31) to one and increments a counter for the number of unexpected CHIPY values. These values should not occur.

8. TIME_RO:

(a) Timed exposure mode:

If

$$\text{DATAMODE} = \text{FAINT or} \quad (207)$$

$$\text{DATAMODE} = \text{FAINT_BIAS or} \quad (208)$$

$$\text{DATAMODE} = \text{GRADED or} \quad (209)$$

$$\text{DATAMODE} = \text{VFAINT}, \quad (210)$$

then

$$\text{TIME_RO} = \text{NaN}. \quad (211)$$

(b) Continuous-clocking mode:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (212)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (213)$$

then

i. Read:

A. Level 0:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT0}, \quad (214)$$

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

B. Level 1, 1.5, or 2:

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (215)$$

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 or} \quad (216)$$

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (217)$$

and

$$\text{TIME_RO}_{\text{in}} > 0, \quad (218)$$

then

$$\text{TIME_RO} = \text{TIME_RO}_{\text{in}}. \quad (219)$$

If

$$\text{CONTENT}_{\text{in}} = \text{EVT1 or} \quad (220)$$

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

$$\text{CONTENT}_{\text{in}} = \text{EVT2} \quad (222)$$

and

$$\text{TIME_RO}_{\text{in}} = 0, \quad (223)$$

then

$$\text{TIME_RO} = \text{TIME}_{\text{in}}. \quad (224)$$

ii. Validation:

If

$$\text{TIME_RO} < 0 \text{ or} \quad (225)$$

$$\text{TIME_RO} \geq 3 \times 10^9, \quad (226)$$

then `acis_process_events` produces a warning upon completion with a count of the total number of events for which either equation 225 or 226 is true. These conditions should not occur.

9. `CHIPX_TARGevt`, `CHIPY_TARGevt`, and `CHIPY_TARGeff`:

The coordinate `CHIPY_TARGeff` is used to compute the coordinates X, Y, and SKY_1D.

(a) Approximate:

If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (227)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (228)$$

and

$$\text{acaofffile} \neq \text{none}, \quad (229)$$

then:

i. Initial values:

The values of `CHIPX_TARGevt` and `CHIPY_TARGevt` are computed from the values of `RA_TARG` and `DEC_TARG` using the value of `CCD_IDfocus` and 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 given by $\text{TIME_RO} - (\text{CHIPY_TARG}_{\text{acaoff,med}} + 1028) \times \text{TIMEDEL}$. The value of `CHIPY_TARGevt` can be negative.

ii. ACIS-I0 or -I2:

If

$$\text{CCD_ID}_{\text{focus}} = 0 \text{ or} \quad (230)$$

$$\text{CCD_ID}_{\text{focus}} = 2, \quad (231)$$

then

A. If

$$\text{CCD_ID} = 0 \text{ or} \quad (232)$$

$$\text{CCD_ID} = 2, \quad (233)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = \text{CHIPY_TARG}_{\text{evt}}. \quad (234)$$

B. If

$$\text{CCD_ID} = 1 \text{ or} \quad (235)$$

$$\text{CCD_ID} = 3, \quad (236)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 - (\text{CHIPY_TARG}_{\text{evt}} - \text{CHIPY_TARG}_{\text{acaoff,med}}). \quad (237)$$

C. If

$$\text{CCD_ID} \geq 4, \quad (238)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 - (\text{CHIPX_TARG}_{\text{evt}} - \text{CHIPX_TARG}_{\text{acaoff,med}}). \quad (239)$$

iii. ACIS-I1 or -I3:

If

$$\text{CCD_ID}_{\text{focus}} = 1 \text{ or} \quad (240)$$

$$\text{CCD_ID}_{\text{focus}} = 3, \quad (241)$$

then

A. If

$$\text{CCD_ID} = 0 \text{ or} \quad (242)$$

$$\text{CCD_ID} = 2, \quad (243)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 - (\text{CHIPY_TARG}_{\text{evt}} - \text{CHIPY_TARG}_{\text{acaoff,med}}). \quad (244)$$

B. If

$$\text{CCD_ID} = 1 \text{ or} \quad (245)$$

$$\text{CCD_ID} = 3, \quad (246)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = \text{CHIPY_TARG}_{\text{evt}}. \quad (247)$$

C. If

$$\text{CCD_ID} \geq 4, \quad (248)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 + (\text{CHIPX_TARG}_{\text{evt}} - \text{CHIPX_TARG}_{\text{acaoff,med}}). \quad (249)$$

iv. ACIS-S:

If

$$\text{CCD_ID}_{\text{focus}} \geq 4, \quad (250)$$

then

A. If

$$\text{CCD_ID} = 0 \text{ or} \quad (251)$$

$$\text{CCD_ID} = 2, \quad (252)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 + (\text{CHIPX_TARG}_{\text{evt}} - \text{CHIPX_TARG}_{\text{acaoff,med}}). \quad (253)$$

B. If

$$\text{CCD_ID} = 1 \text{ or} \quad (254)$$

$$\text{CCD_ID} = 3, \quad (255)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = 512 - (\text{CHIPX_TARG}_{\text{evt}} - \text{CHIPX_TARG}_{\text{acaoff,med}}). \quad (256)$$

C. If

$$\text{CCD_ID} \geq 4, \quad (257)$$

then

$$\text{CHIPY_TARG}_{\text{eff}} = \text{CHIPY_TARG}_{\text{evt}}. \quad (258)$$

(b) Validation:

If

$$\text{CHIPY_TARG}_{\text{eff}} < -256 \text{ or} \quad (259)$$

$$\text{CHIPY_TARG}_{\text{eff}} \geq 1280, \quad (260)$$

then `acis_process_events` produces a warning message.

10. TG_M:

(a) Continuous-clocking mode with gratings:

If

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \quad (261)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (262)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (263)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (264)$$

then

- i. Read:
The value of `TG_M` for an event is given by `TG_Min`.
- ii. Validation:

A. If

$$\text{TG_M} < -99, \quad (265)$$

then

$$\text{TG_M} = -99 \quad (266)$$

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

$$\text{TG_M} > 99, \quad (267)$$

then

$$\text{TG_M} = 99 \quad (268)$$

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.

11. CHIPY_TG:

- (a) Continuous-clocking mode with gratings:

If

$$\text{OBS_MODE}_{\text{in}} = \text{pointing} \quad (269)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (270)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (271)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1}, \quad (272)$$

then

- i. Read:
The value of `CHIPY_TG` for an event is given by `CHIPY_TGin`.
- ii. Validation:

A. If

$$\text{TG_M} > -99 \text{ and} \quad (273)$$

$$\text{TG_M} < 99 \text{ and} \quad (274)$$

$$\text{CHIPY_TG} \neq \text{NaN} \quad (275)$$

and

$$\text{CHIPY_TG} \leq 0 \text{ or} \quad (276)$$

$$\text{CHIPY_TG} \geq 1025, \quad (277)$$

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

$$\text{TG_M} > -99 \text{ and} \quad (278)$$

$$\text{TG_M} < 99 \text{ and} \quad (279)$$

$$\text{CHIPY_TG} \neq \text{NaN} \text{ and} \quad (280)$$

$$\text{CHIPY_TG} < 1, \quad (281)$$

then

$$\text{CHIPY_TG} = 1. \quad (282)$$

C. If

$$\text{TG_M} > -99 \text{ and} \quad (283)$$

$$\text{TG_M} < 99 \text{ and} \quad (284)$$

$$\text{CHIPY_TG} \neq \text{NaN} \text{ and} \quad (285)$$

$$\text{CHIPY_TG} > 1024, \quad (286)$$

then

$$\text{CHIPY_TG} = 1024. \quad (287)$$

12. TIME and CHIPY_ADJ:

For continuous-clocking mode observations, the value of CHIPY_ADJ is used to compute the TIME, pulse heights, and the coordinates TDET_X, TDET_Y, DET_X, and DET_Y.

(a) Calculate:

i. Timed exposure mode:

If

$$\text{DATAMODE}_{i_n} = \text{FAINT} \text{ or} \quad (288)$$

$$\text{DATAMODE}_{i_n} = \text{FAINT_BIAS} \text{ or} \quad (289)$$

$$\text{DATAMODE}_{i_n} = \text{GRADED} \text{ or} \quad (290)$$

$$\text{DATAMODE}_{i_n} = \text{VFAINT}, \quad (291)$$

then

$$\text{TIME} = \text{TIME}_{i_n} \text{ and} \quad (292)$$

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

ii. Continuous-clocking mode:

If

$$\text{DATAMODE}_{i_n} = \text{CC33_FAINT} \text{ or} \quad (294)$$

$$\text{DATAMODE}_{i_n} = \text{CC33_GRADED} \quad (295)$$

then

A. Set

$$\text{CHIPY_ADJ} = 512 \text{ and} \quad (296)$$

$$\text{TIME}' = \text{TIME_RO} - (\text{CHIPY_ADJ} + 1028) \times \text{TIMEDEL}. \quad (297)$$

B. If

$$\text{acaofffile} \neq \text{none}, \quad (298)$$

then

$$\text{CHIPY_ADJ}' = \text{CHIPY_TARG}_{\text{eff}} \text{ and} \quad (299)$$

$$\text{TIME}' = \text{TIME_RO} - (\text{CHIPY_ADJ}' + 1028) \times \text{TIMEDEL}, \quad (300)$$

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

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ and} \quad (301)$$

$$\text{TG_M} > -99 \text{ and} \quad (302)$$

$$\text{TG_M} < 99 \text{ and} \quad (303)$$

$$\text{CHIPY_TG} \neq \text{NaN}, \quad (304)$$

then

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

D. If

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \quad (306)$$

and

$$\text{TG_M} = -99 \text{ or} \quad (307)$$

$$\text{TG_M} = 99, \quad (308)$$

then

$$\text{CHIPY_ADJ} = 512. \quad (309)$$

E. If

$$\text{acaofffile} \neq \text{none} \quad (310)$$

and

$$\text{TIME}' < \text{TIME}_{\text{min}} \text{ or} \quad (311)$$

$$\text{TIME}' \geq \text{TIME}_{\text{max}}, \quad (312)$$

then

$$\text{CHIPY_ADJ} = 512. \quad (313)$$

F. If

$$\text{acaofffile} \neq \text{none} \quad (314)$$

and

$$\text{TIME}' < \text{TIME}_{\text{min}} \text{ or} \quad (315)$$

$$\text{TIME}' \geq \text{TIME}_{\text{max}}, \quad (316)$$

and

$$\text{CCD_ID}_{\text{focus}} = 0 \text{ or} \quad (317)$$

$$\text{CCD_ID}_{\text{focus}} = 2 \quad (318)$$

and

$$\text{CCD_ID} = 0 \text{ or} \quad (319)$$

$$\text{CCD_ID} = 2, \quad (320)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_TARG}_{\text{acaoff,med}}. \quad (321)$$

G. If

$$\text{acaofffile} \neq \text{none} \quad (322)$$

and

$$\text{TIME}' < \text{TIME}_{\text{min}} \text{ or} \quad (323)$$

$$\text{TIME}' \geq \text{TIME}_{\text{max}}, \quad (324)$$

and

$$\text{CCD_ID}_{\text{focus}} = 1 \text{ or} \quad (325)$$

$$\text{CCD_ID}_{\text{focus}} = 3 \quad (326)$$

and

$$\text{CCD_ID} = 1 \text{ or} \quad (327)$$

$$\text{CCD_ID} = 3, \quad (328)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_TARG}_{\text{acaoff,med}}. \quad (329)$$

H. If

$$\text{acaofffile} \neq \text{none} \quad (330)$$

and

$$\text{TIME}' < \text{TIME}_{\text{min}} \text{ or} \quad (331)$$

$$\text{TIME}' \geq \text{TIME}_{\text{max}}, \quad (332)$$

and

$$\text{CCD_ID}_{\text{focus}} \geq 4 \text{ and} \quad (333)$$

$$\text{CCD_ID} \geq 4, \quad (334)$$

then

$$\text{CHIPY_ADJ} = \text{CHIPY_TARG}_{\text{acaoff,med}}. \quad (335)$$

I. If $\text{acaofffile} = \text{none}$ (336)

and

$\text{CONTENT}_{\text{in}} \neq \text{TGEVT1}$, (337)

then

$\text{CHIPY_ADJ} = 512$. (338)

J. If

$\text{CCD_ID}_{\text{focus}} \leq 3$ and (339)

$\text{CCD_ID} \geq 4$, (340)

then

$\text{CHIPY_ADJ} = 512$. (341)

K. If

$\text{CCD_ID}_{\text{focus}} \geq 4$ and (342)

$\text{CCD_ID} \leq 3$, (343)

then

$\text{CHIPY_ADJ} = 512$. (344)

L. If

$\text{OBS_MODE}_{\text{in}} = \text{secondary}$, (345)

then

$\text{CHIPY_ADJ} = 512$. (346)

M. If

$\text{CHIPY_ADJ} < 0.5$ or (347)

$\text{CHIPY_ADJ} \geq 1024.5$ (348)

then

$\text{CHIPY_ADJ} = 512$. (349)

N. Set

$\text{TIME} = \text{TIME_RO} - (\text{CHIPY_ADJ} + 1028) \times \text{TIMEDEL}$. (350)

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

$$\text{TIME} < 0 \text{ or} \quad (351)$$

$$\text{TIME} \geq 3 \times 10^9, \quad (352)$$

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

$$\text{CHIPY_ADJ} < 0.5 \text{ or} \quad (353)$$

$$\text{CHIPY_ADJ} \geq 1024.5, \quad (354)$$

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

$$\text{badpixfile} \neq \text{none} \quad (355)$$

and 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 there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (356)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (357)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (358)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (359)$$

and

$$\text{STATUS}_{\text{badpix},r}[5] = 1 \text{ or} \quad (360)$$

$$\text{STATUS}_{\text{badpix},r}[6] = 1 \text{ or} \quad (361)$$

$$\text{STATUS}_{\text{badpix},r}[9] = 1, \quad (362)$$

then

$$\text{STATUS}[0] = 1 \quad (363)$$

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 there are one or more rows r in HDU h_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (364)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (365)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (366)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (367)$$

and

$$\text{STATUS}_{\text{badpix},r}[0] = 1 \text{ or} \quad (368)$$

$$\text{STATUS}_{\text{badpix},r}[1] = 1 \text{ or} \quad (369)$$

$$\text{STATUS}_{\text{badpix},r}[7] = 1 \text{ or} \quad (370)$$

$$\text{STATUS}_{\text{badpix},r}[11] = 1 \text{ or} \quad (371)$$

$$\text{STATUS}_{\text{badpix},r}[12] = 1 \text{ or} \quad (372)$$

$$\text{STATUS}_{\text{badpix},r}[13] = 1 \text{ or} \quad (373)$$

$$\text{STATUS}_{\text{badpix},r}[14] = 1 \text{ or} \quad (374)$$

$$\text{STATUS}_{\text{badpix},r}[16] = 1, \quad (375)$$

then

$$\text{STATUS}[4] = 1 \quad (376)$$

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_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (377)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (378)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (379)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (380)$$

and

$$\text{STATUS}_{\text{badpix},r}[8] = 1 \text{ or} \quad (381)$$

$$\text{STATUS}_{\text{badpix},r}[10] = 1, \quad (382)$$

then

$$\text{STATUS}[5] = 1 \quad (383)$$

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_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (384)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (385)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (386)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (387)$$

and

$$\text{STATUS}_{\text{badpix},r}[3] = 1, \quad (388)$$

then

$$\text{STATUS}[6] = 1 \quad (389)$$

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_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (390)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (391)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (392)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (393)$$

and

$$\text{STATUS}_{\text{badpix},r}[2] = 1 \text{ or} \quad (394)$$

$$\text{STATUS}_{\text{badpix},r}[4] = 1, \quad (395)$$

then

$$\text{STATUS}[8] = 1 \quad (396)$$

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_{badpix} where

$$\text{CHIPX} \geq \text{CHIPX}_{\text{badpix},r}[0] \text{ and} \quad (397)$$

$$\text{CHIPX} \leq \text{CHIPX}_{\text{badpix},r}[1] \text{ and} \quad (398)$$

$$\text{TIME} \geq \text{TIME}_{\text{badpix},r} \text{ and} \quad (399)$$

$$\text{TIME} < \text{TIME_STOP}_{\text{badpix},r} \quad (400)$$

and

$$\text{STATUS}_{\text{badpix},r}[15] = 1, \quad (401)$$

then

$$\text{STATUS}[16] = 1 \quad (402)$$

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

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. `FLTGRADE_RO`:

- (a) If HDU h_{in} of the `infile` includes the column `PHAS` and

$$\text{doevtgrade} = \text{yes}, \quad (403)$$

then

$$\text{FLTGRADE_RO} = c_{\text{fro}}[0] + 2c_{\text{fro}}[1] + 4c_{\text{fro}}[2] + 8c_{\text{fro}}[3] + \quad (404)$$

$$16c_{\text{fro}}[5] + 32c_{\text{fro}}[6] + 64c_{\text{fro}}[7] + 128c_{\text{fro}}[8]. \quad (405)$$

The values of $c_{\text{fro}}[j]$ are determined as follows.

- i. Each value of $c_{\text{fro}}[j]$ is initialized such that

$$c_{\text{fro}}[j] = 1. \quad (406)$$

- ii. If

$$\text{PHAS}[j] < \text{split threshold or} \quad (407)$$

$$\text{PHAS}[j] > 4095, \quad (408)$$

then

$$c_{\text{fro}}[j] = 0. \quad (409)$$

- iii. If

$$j \geq 0 \text{ and} \quad (410)$$

$$j \leq 3 \text{ and} \quad (411)$$

$$\text{PHAS}[j] > \text{PHAS}[4], \quad (412)$$

then

$$c_{\text{fro}}[j] = 0. \quad (413)$$

- iv. If

$$j \geq 5 \text{ and} \quad (414)$$

$$j \leq 8 \text{ and} \quad (415)$$

$$\text{PHAS}[j] \geq \text{PHAS}[4], \quad (416)$$

then

$$c_{\text{fro}}[j] = 0. \quad (417)$$

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

- (b) If HDU h_{in} of the `infile` does not include the column `PHAS` or

$$\text{doevtgrade} = \text{no}, \quad (418)$$

then

$$\text{FLTGRADE_RO} = \text{NULL}. \quad (419)$$

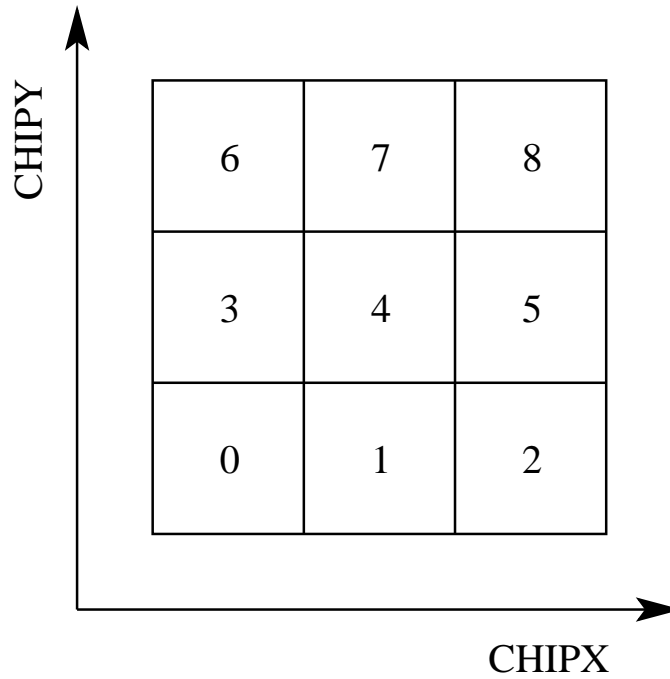


Figure 1: The relative CHIPX and CHIPY coordinates of the nine elements $j = 0-8$ of a 3 pixel \times 3 pixel event island PHAS[j] or PHAS_ADJ[j].

16. GRADE_RO:

(a) If

$$\text{FLTGRADE_RO} \neq \text{NULL} \text{ and} \tag{420}$$

$$\text{gradefile} \neq \text{none} \text{ and} \tag{421}$$

HDU h_{grade} of the `gradefile` has a row r , where

$$\text{FLTGRADE}_{\text{grade}}[r] = \text{FLTGRADE_RO}, \tag{422}$$

then

$$\text{GRADE_RO} = \text{GRADE}_{\text{grade}}[r]. \tag{423}$$

(b) If

$$\text{FLTGRADE_RO} = \text{NULL} \text{ or} \tag{424}$$

$$\text{gradefile} = \text{none} \text{ or} \tag{425}$$

HDU h_{grade} of the `gradefile` does not have a row r , where

$$\text{FLTGRADE}_{\text{grade}}[r] = \text{FLTGRADE_RO}, \tag{426}$$

then

$$\text{GRADE_RO} = \text{NULL}. \tag{427}$$

17. PHA_RO:

(a) If HDU h_{in} of the `infile` includes the column PHAS and

$$\text{doevtgrade} = \text{yes and} \quad (428)$$

$$\text{GRADE_RO} \neq \text{NULL}, \quad (429)$$

then

$$\text{PHA_RO} = \sum_{j=0}^8 c_{\text{pro}}[j] \text{PHAS}[j]. \quad (430)$$

The values of $c_{\text{pro}}[j]$ are determined as follows.

i. Each value of $c_{\text{pro}}[j]$ is initialized such that

$$c_{\text{pro}}[j] = 1. \quad (431)$$

ii. If

$$\text{PHAS}[j] < \text{split threshold}, \quad (432)$$

then

$$c_{\text{pro}}[j] = 0. \quad (433)$$

iii. If

$$j \geq 0 \text{ and} \quad (434)$$

$$j \leq 3 \text{ and} \quad (435)$$

$$\text{PHAS}[j] > \text{PHAS}[4], \quad (436)$$

then

$$c_{\text{pro}}[j] = 0. \quad (437)$$

iv. If

$$j \geq 5 \text{ and} \quad (438)$$

$$j \leq 8 \text{ and} \quad (439)$$

$$\text{PHAS}[j] \geq \text{PHAS}[4], \quad (440)$$

then

$$c_{\text{pro}}[j] = 0. \quad (441)$$

v. If

$$\text{CORNERS} = -1, \quad (442)$$

then

$$c_{\text{pro}}[0] = 0 \text{ and} \quad (443)$$

$$c_{\text{pro}}[2] = 0 \text{ and} \quad (444)$$

$$c_{\text{pro}}[6] = 0 \text{ and} \quad (445)$$

$$c_{\text{pro}}[8] = 0. \quad (446)$$

vi. If

$$\text{CORNERS} = 0, \quad (447)$$

then there are no additional constraints on $c_{\text{pro}}[0]$, $c_{\text{pro}}[2]$, $c_{\text{pro}}[6]$, and $c_{\text{pro}}[8]$.

vii. If

$$\text{CORNERS} = 1, \quad (448)$$

then

A. If

$$c_{\text{pro}}[1] = 0 \text{ and} \quad (449)$$

$$c_{\text{pro}}[3] = 0, \quad (450)$$

then

$$c_{\text{pro}}[0] = 0. \quad (451)$$

B. If

$$c_{\text{pro}}[1] = 0 \text{ and} \quad (452)$$

$$c_{\text{pro}}[5] = 0, \quad (453)$$

then

$$c_{\text{pro}}[2] = 0. \quad (454)$$

C. If

$$c_{\text{pro}}[3] = 0 \text{ and} \quad (455)$$

$$c_{\text{pro}}[7] = 0, \quad (456)$$

then

$$c_{\text{pro}}[6] = 0. \quad (457)$$

D. If

$$c_{\text{pro}}[5] = 0 \text{ and} \quad (458)$$

$$c_{\text{pro}}[7] = 0, \quad (459)$$

then

$$c_{\text{pro}}[8] = 0. \quad (460)$$

viii. If

$$\text{CORNERS} = 2, \quad (461)$$

then

A. If

$$c_{\text{pro}}[1] = 0 \text{ or} \quad (462)$$

$$c_{\text{pro}}[3] = 0 \text{ or} \quad (463)$$

$$\text{GRADE_RO} \neq 6, \quad (464)$$

then

$$c_{\text{pro}}[0] = 0. \quad (465)$$

B. If

$$c_{\text{pro}}[1] = 0 \text{ or} \quad (466)$$

$$c_{\text{pro}}[5] = 0 \text{ or} \quad (467)$$

$$\text{GRADE_RO} \neq 6, \quad (468)$$

then

$$c_{\text{pro}}[2] = 0. \quad (469)$$

C. If

$$c_{\text{pro}}[3] = 0 \text{ or} \quad (470)$$

$$c_{\text{pro}}[7] = 0 \text{ or} \quad (471)$$

$$\text{GRADE_RO} \neq 6, \quad (472)$$

then

$$c_{\text{pro}}[6] = 0. \quad (473)$$

D. If

$$c_{\text{pro}}[5] = 0 \text{ or} \quad (474)$$

$$c_{\text{pro}}[7] = 0 \text{ or} \quad (475)$$

$$\text{GRADE_RO} \neq 6, \quad (476)$$

then

$$c_{\text{pro}}[8] = 0. \quad (477)$$

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

(b) If HDU h_{in} of the `infile` does not include the column PHAS or

$$\text{doevtgrade} = \text{no or} \quad (478)$$

$$\text{GRADE_RO} = \text{NULL}, \quad (479)$$

then

i. If HDU h_{in} of the `infile` includes the column PHA_RO, then

$$\text{PHA_RO} = \text{PHA_RO}_{\text{in}}. \quad (480)$$

ii. If HDU h_{in} of the `infile` does not include the column PHA_RO, then

A. If HDU h_{in} of the `infile` includes the column PHA and HDU h_{in} of the `infile` includes the keyword TGAINCOR and

$$\text{TGAINCOR} = 0, \quad (481)$$

then

$$\text{PHA_RO} = \text{PHA}_{\text{in}}. \quad (482)$$

B. If HDU h_{in} of the `infile` includes the column PHA and HDU h_{in} of the `infile` includes the keyword TGAINCOR and

$$\text{TGAINCOR} \neq 0, \quad (483)$$

then

$$\text{PHA_RO} = \text{NULL}. \quad (484)$$

- C. If HDU h_{in} of the `infile` includes the column `PHA` and HDU h_{in} of the `infile` does not include the keyword `TGAINCOR` then

$$\text{PHA_RO} = \text{PHA}_{\text{in}}. \quad (485)$$

- D. If HDU h_{in} of the `infile` does not include the column `PHA`, then

$$\text{PHA_RO} = \text{NULL}. \quad (486)$$

18. 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 Δ_x , the parallel CTI adjustment Δ_y , and the adjusted pulse heights `PHAS_ADJ` are initialized such that

$$\Delta_x[j] = 0, \quad (487)$$

$$\Delta_y[j] = 0, \text{ and} \quad (488)$$

$$\text{PHAS_ADJ}[j] = \text{PHAS}[j] \quad (489)$$

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

$$n = 1. \quad (490)$$

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

$$\Delta'_x[j] = \Delta_x[j], \quad (491)$$

$$\Delta'_y[j] = \Delta_y[j], \text{ and} \quad (492)$$

$$\text{PHAS_ADJ}'[j] = \text{PHAS_ADJ}[j] \quad (493)$$

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 Δ_x are given by

$$\Delta_x[0] = c_x[0]s_x\rho_x[0]V_x[0], \quad (494)$$

$$\Delta_x[1] = c_x[1]s_x\rho_x[1]V_x[1] - c'_x[0]s_x\rho_x[0]V_x[0], \quad (495)$$

$$\Delta_x[2] = c_x[2]s_x\rho_x[2]V_x[2] - c'_x[1]s_x\rho_x[1]V_x[1], \quad (496)$$

$$\Delta_x[3] = c_x[3]s_x\rho_x[3]V_x[3], \quad (497)$$

$$\Delta_x[4] = c_x[4]s_x\rho_x[4]V_x[4] - c'_x[3]s_x\rho_x[3]V_x[3], \quad (498)$$

$$\Delta_x[5] = c_x[5]s_x\rho_x[5]V_x[5] - c'_x[4]s_x\rho_x[4]V_x[4], \quad (499)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6], \quad (500)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[6]s_x\rho_x[6]V_x[6], \text{ and} \quad (501)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8] - c'_x[7]s_x\rho_x[7]V_x[7], \quad (502)$$

$$\rho_x[j] = \begin{cases} \text{TIME}_{k+1} \text{ is the } (k+1)^{\text{th}} \text{ element of the column TIME in the mtlfile,} \\ t'_{k+1} > t', \\ \text{If } t' > t'_k \text{ for } k = n, \text{ where } n \text{ is the last element, then } k = n, \\ \text{FP_TEMP}_{k+1} \text{ is the } (k+1)^{\text{th}} \text{ element of the column FP_TEMP in the} \\ \text{mtlfile,} \end{cases} \quad (508)$$

$$V_x[j] = \begin{cases} \rho_x[j] \text{ depends upon the CCD_ID and upon the CHIPX and nint(CHIPY_ADJ)} \\ \text{coordinates associated with element } j \text{ of PHAS_ADJ}[j] \text{ (see Fig. 1),} \\ \left(\frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME_X}_{l+1} - \text{VOLUME_X}_l) + \\ \text{VOLUME_X}_l, \end{cases} \quad (509)$$

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

- 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 Δ_x are given by

$$\Delta_x[0] = c_x[0]s_x\rho_x[0]V_x[0] - c'_x[1]s_x\rho_x[1]V_x[1], \quad (510)$$

$$\Delta_x[1] = c_x[1]s_x\rho_x[1]V_x[1] - c'_x[2]s_x\rho_x[2]V_x[2], \quad (511)$$

$$\Delta_x[2] = c_x[2]s_x\rho_x[2]V_x[2], \quad (512)$$

$$\Delta_x[3] = c_x[3]s_x\rho_x[3]V_x[3] - c'_x[4]s_x\rho_x[4]V_x[4], \quad (513)$$

$$\Delta_x[4] = c_x[4]s_x\rho_x[4]V_x[4] - c'_x[5]s_x\rho_x[5]V_x[5], \quad (514)$$

$$\Delta_x[5] = c_x[5]s_x\rho_x[5]V_x[5], \quad (515)$$

$$\Delta_x[6] = c_x[6]s_x\rho_x[6]V_x[6] - c'_x[7]s_x\rho_x[7]V_x[7], \quad (516)$$

$$\Delta_x[7] = c_x[7]s_x\rho_x[7]V_x[7] - c'_x[8]s_x\rho_x[8]V_x[8], \text{ and} \quad (517)$$

$$\Delta_x[8] = c_x[8]s_x\rho_x[8]V_x[8], \quad (518)$$

where

$$\begin{aligned}
c_x[j] &= \left\{ \begin{array}{l} 0 \\ \text{FRCTRLX} \\ 1 \end{array} \right. \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j\text{)}, \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \\ \text{(for } j = 0, 1, 3, 4, 6, 7\text{)}, \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 2, 5, 8) \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j+1] + \Delta'_x[j+1] + \Delta'_y[j+1] \\ \text{(for } j = 0, 1, 3, 4, 6, 7\text{)}, \end{array} \right. \\
c'_x[j] &= \left\{ \begin{array}{l} 0 \\ \text{FRCTRLX} \\ 1 \end{array} \right. \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] < \text{split threshold or} \\ j \rightarrow \text{CHIPX} = 257, 512, 769, \text{ or } 1024 \\ \text{(for } j = 1, 2, 4, 5, 7, 8\text{)}, \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \text{ and} \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \geq \text{split threshold} \\ \text{(for } j = 1, 2, 4, 5, 7, 8\text{)}, \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j-1] + \Delta'_x[j-1] + \Delta'_y[j-1] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 1, 2, 4, 5, 7, 8\text{)}, \end{array} \right.
\end{aligned}$$

and s_x , T , t' , t'_k , t'_{k+1} , $\rho_x[j]$, and $V_x[j]$ are given by equations. 503, 504, 505, 506, 507, 508, and 509, 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

$$\Delta_y[0] = c_y[0]s_y\rho_y[0]V_y[0], \quad (519)$$

$$\Delta_y[1] = c_y[1]s_y\rho_y[1]V_y[1], \quad (520)$$

$$\Delta_y[2] = c_y[2]s_y\rho_y[2]V_y[2], \quad (521)$$

$$\Delta_y[3] = c_y[3]s_y\rho_y[3]V_y[3] - c'_y[0]s_y\rho_y[0]V_y[0], \quad (522)$$

$$\Delta_y[4] = c_y[4]s_y\rho_y[4]V_y[4] - c'_y[1]s_y\rho_y[1]V_y[1], \quad (523)$$

$$\Delta_y[5] = c_y[5]s_y\rho_y[5]V_y[5] - c'_y[2]s_y\rho_y[2]V_y[2], \quad (524)$$

$$\Delta_y[6] = c_y[6]s_y\rho_y[6]V_y[6] - c'_y[3]s_y\rho_y[3]V_y[3], \quad (525)$$

$$\Delta_y[7] = c_y[7]s_y\rho_y[7]V_y[7] - c'_y[4]s_y\rho_y[4]V_y[4], \text{ and} \quad (526)$$

$$\Delta_y[8] = c_y[8]s_y\rho_y[8]V_y[8] - c'_y[5]s_y\rho_y[5]V_y[5], \quad (527)$$

where

$$\begin{aligned}
c_y[j] &= \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold} \\ \text{(for all } j\text{)}, \\ \text{FRCTRLY} \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8\text{)}, \end{array} \right. \\ \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2\text{)} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \\ \text{PHAS}[j-3] + \Delta'_x[j-3] + \Delta'_y[j-3] \\ \text{(for } j = 3, 4, 5, 6, 7, 8\text{)}, \end{array} \right. \end{cases} \\
c'_y[j] &= \begin{cases} 0 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] < \text{split threshold or} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] < \text{split threshold or} \\ j \rightarrow \text{CHIPY} = 1 \text{ or } 1024 \\ \text{(for } j = 1, 2, 3, 4, 5\text{)}, \\ \text{FRCTRLY} \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] > \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5\text{)}, \end{array} \right. \\ \\ 1 & \left\{ \begin{array}{l} \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \leq \\ \text{PHAS}[j+3] + \Delta'_x[j+3] + \Delta'_y[j+3] \text{ and} \\ \text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] \geq \text{split threshold} \\ \text{(for } j = 0, 1, 2, 3, 4, 5\text{)}, \end{array} \right. \end{cases} \\
s_y &= 1 + \text{TCTIY}(T - \text{FP_TEMPO}), \tag{528}
\end{aligned}$$

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

$$\begin{aligned}
\rho_y[j] &= \text{parallel trap density,} \tag{529} \\
&\left\{ \begin{array}{l} \rho_y[j] \text{ depends upon the CCD_ID and upon the CHIPX and nint(CHIPY_ADJ)} \\ \text{coordinates associated with element } j \text{ of PHAS_ADJ}[j] \text{ (see Fig. 1),} \end{array} \right.
\end{aligned}$$

$$\begin{aligned}
V_y[j] &= \left(\frac{\text{PHAS}[j] + \Delta'_x[j] + \Delta'_y[j] - \text{PHA}_l}{\text{PHA}_{l+1} - \text{PHA}_l} \right) (\text{VOLUME_Y}_{l+1} - \text{VOLUME_Y}_l) + \\
&\text{VOLUME_Y}_l, \tag{530}
\end{aligned}$$

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

and T , t' , t'_k , and t'_{k+1} , are given by equations. 504, 505, 506, and 507, respectively.
vi. The CTI-adjusted pulse heights

$$\text{PHAS_ADJ}[j] = \text{PHAS}[j] + \Delta_x[j] + \Delta_y[j] \quad (531)$$

for all j .

vii. A. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| < \text{cticonverge} \text{ (for all } j \text{) and} \quad (532)$$

$$n \leq \text{max_cti_iter}, \quad (533)$$

then the computation of PHAS_ADJ is complete for the event.

B. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| \geq \text{cticonverge} \text{ (for one or more } j \text{) and} \quad (534)$$

$$n < \text{max_cti_iter}, \quad (535)$$

then $n = n + 1$ and steps 1.5.18(a)iii–1.5.18(a)vii are repeated.

C. If

$$|\text{PHAS_ADJ}'[j] - \text{PHAS_ADJ}[j]| \geq \text{cticonverge} \text{ (for one or more } j \text{) and} \quad (536)$$

$$n \geq \text{max_cti_iter}, \quad (537)$$

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.

19. FLTGRADE:

(a) If HDU h_{in} of the `infile` includes the column PHAS and

$$\text{apply_cti} = \text{yes and} \quad (538)$$

$$\text{doevtgrade} = \text{yes}, \quad (539)$$

then

$$\text{FLTGRADE} = c_f[0] + 2c_f[1] + 4c_f[2] + 8c_f[3] + \quad (540)$$

$$16c_f[5] + 32c_f[6] + 64c_f[7] + 128c_f[8]. \quad (541)$$

The values of $c_f[j]$ are determined as follows.

i. Each value of $c_f[j]$ is initialized such that

$$c_f[j] = 1. \quad (542)$$

ii. If

$$\text{PHAS_ADJ}[j] < \text{split threshold}, \quad (543)$$

then

$$c_f[j] = 0. \quad (544)$$

The elements $j = 0$ –8 of PHAS_ADJ are depicted in Figure 1.

(b) If HDU h_{in} of the `infile` includes the column `PHAS` and

$$\text{apply_cti} = \text{no and} \quad (545)$$

$$\text{doevtgrade} = \text{yes,} \quad (546)$$

then

$$\text{FLTGRADE} = \text{FLTGRADE_RO.} \quad (547)$$

(c) If HDU h_{in} of the `infile` does not include the column `PHAS` or

$$\text{doevtgrade} = \text{no,} \quad (548)$$

then

i. If HDU h_{in} of the `infile` includes the column `FLTGRADE`, then

$$\text{FLTGRADE} = \text{FLTGRADE}_{\text{in}}. \quad (549)$$

ii. If HDU h_{in} of the `infile` does not include the column `FLTGRADE`, then

$$\text{FLTGRADE} = \text{NULL.} \quad (550)$$

20. `GRADE`:

(a) If

$$\text{FLTGRADE} \neq \text{NULL and} \quad (551)$$

$$\text{gradefile} \neq \text{none and} \quad (552)$$

HDU h_{grade} of the `gradefile` has a row r , where

$$\text{FLTGRADE}_{\text{grade}}[r] = \text{FLTGRADE,} \quad (553)$$

then

$$\text{GRADE} = \text{GRADE}_{\text{grade}}[r]. \quad (554)$$

(b) If

$$\text{FLTGRADE} = \text{NULL or} \quad (555)$$

$$\text{gradefile} = \text{none or} \quad (556)$$

HDU h_{grade} of the `gradefile` does not have a row r , where

$$\text{FLTGRADE}_{\text{grade}}[r] = \text{FLTGRADE,} \quad (557)$$

then

$$\text{GRADE} = \text{NULL.} \quad (558)$$

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

(a) If HDU h_{in} of the `infile` includes the column `PHAS` and

$$\text{apply_cti} = \text{yes and} \quad (559)$$

$$\text{doevtgrade} = \text{yes and} \quad (560)$$

$$\text{GRADE} \neq \text{NULL}, \quad (561)$$

then

$$\text{PHA} = \sum_{j=0}^8 c_p[j] \text{PHAS_ADJ}[j]. \quad (562)$$

The values of $c_p[j]$ are determined as follows.

i. Each value of $c_p[j]$ is initialized such that

$$c_p[j] = 1. \quad (563)$$

ii. If

$$\text{PHAS_ADJ}[j] < \text{split threshold}, \quad (564)$$

then

$$c_p[j] = 0. \quad (565)$$

iii. If

$$\text{CORNERS} = -1, \quad (566)$$

then

$$c_p[0] = 0 \text{ and} \quad (567)$$

$$c_p[2] = 0 \text{ and} \quad (568)$$

$$c_p[6] = 0 \text{ and} \quad (569)$$

$$c_p[8] = 0. \quad (570)$$

iv. If

$$\text{CORNERS} = 0, \quad (571)$$

then there are no additional constraints on $c_p[0]$, $c_p[2]$, $c_p[6]$, and $c_p[8]$.

v. If

$$\text{CORNERS} = 1, \quad (572)$$

then

A. If

$$c_p[1] = 0 \text{ and} \quad (573)$$

$$c_p[3] = 0, \quad (574)$$

then

$$c_p[0] = 0. \quad (575)$$

B. If

$$c_p[1] = 0 \text{ and} \quad (576)$$

$$c_p[5] = 0, \quad (577)$$

then

$$c_p[2] = 0. \quad (578)$$

C. If

$$c_p[3] = 0 \text{ and} \quad (579)$$

$$c_p[7] = 0, \quad (580)$$

then

$$c_p[6] = 0. \quad (581)$$

D. If

$$c_p[5] = 0 \text{ and} \quad (582)$$

$$c_p[7] = 0, \quad (583)$$

then

$$c_p[8] = 0. \quad (584)$$

vi. If

$$\text{CORNERS} = 2, \quad (585)$$

then

A. If

$$c_p[1] = 0 \text{ or} \quad (586)$$

$$c_p[3] = 0 \text{ or} \quad (587)$$

$$\text{GRADE} \neq 6, \quad (588)$$

then

$$c_p[0] = 0. \quad (589)$$

B. If

$$c_p[1] = 0 \text{ or} \quad (590)$$

$$c_p[5] = 0 \text{ or} \quad (591)$$

$$\text{GRADE} \neq 6, \quad (592)$$

then

$$c_p[2] = 0. \quad (593)$$

C. If

$$c_p[3] = 0 \text{ or} \quad (594)$$

$$c_p[7] = 0 \text{ or} \quad (595)$$

$$\text{GRADE} \neq 6, \quad (596)$$

then

$$c_p[6] = 0. \quad (597)$$

D. If

$$c_p[5] = 0 \text{ or} \quad (598)$$

$$c_p[7] = 0 \text{ or} \quad (599)$$

$$\text{GRADE} \neq 6, \quad (600)$$

then

$$c_p[8] = 0. \quad (601)$$

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

(b) If HDU h_{in} of the `infile` does not include the column PHAS or

$$\text{apply_cti} = \text{no or} \quad (602)$$

$$\text{doevtgrade} = \text{no or} \quad (603)$$

$$\text{GRADE} = \text{NULL}, \quad (604)$$

then

$$\text{PHA} = \text{PHA_R0}. \quad (605)$$

(c) If

$$\text{apply_tgain} = \text{yes}, \quad (606)$$

then

$$\text{PHA} = \text{PHA} - \text{int} \left[\left(\frac{\text{TIME} - \text{EPOCH1}}{\text{EPOCH2} - \text{EPOCH1}} \right) (\delta_2 - \delta_1) + \delta_1 - \epsilon \right], \quad (607)$$

where

$$\text{int} = \text{the integer portion of (i.e. truncate or round down)}, \quad (608)$$

$$\text{TIME} = \text{the time of the event}, \quad (609)$$

$$\text{EPOCH1} = \text{a keyword in the } \text{tgainfile}, \quad (610)$$

$$\text{EPOCH2} = \text{a keyword in the } \text{tgainfile}, \quad (611)$$

$$\delta_1 = \left(\frac{\text{PHA} - \text{PHA}_m[r]}{\text{PHA}_{m+1}[r] - \text{PHA}_m[r]} \right) (\text{DELTPHA1}_{m+1}[r] - \text{DELTPHA1}_m[r]) + \quad (612)$$

$$\text{DELTPHA1}_m[r], \quad (613)$$

$$\left\{ \begin{array}{l}
r \text{ is the row of the } \mathbf{tgainfile} \text{ where} \\
\left\{ \begin{array}{l}
\text{CCD_ID}[r] = \text{CCD_ID}, \\
\text{CHIPX_LO}[r] \leq \text{CHIPX}, \\
\text{CHIPX_HI}[r] \geq \text{CHIPX}, \\
\text{CHIPY_LO}[r] \leq \text{nint}(\text{CHIPY_ADJ}), \text{ and} \\
\text{CHIPY_HI}[r] \geq \text{nint}(\text{CHIPY_ADJ}).
\end{array} \right. \\
m \text{ is the element of row } r \text{ where} \\
\left\{ \begin{array}{l}
\text{PHA}_m[r] \leq \text{PHA} \text{ and} \\
\text{PHA}_{m+1}[r] > \text{PHA}. \\
\text{If } \text{PHA} < \text{PHA}_m[r] \text{ for } m = 0, \text{ then } m = 0. \\
\text{If } \text{PHA} \geq \text{PHA}_m[r] \text{ for } m = M \text{ and } M \text{ is the last element of } \text{PHA}[r], \\
\text{then } m = M - 1.
\end{array} \right. \\
\text{The } \mathbf{tgainfile} \text{ includes a binary table with columns named} \\
\text{CCD_ID, CHIPX_LO, CHIPX_HI, CHIPY_LO, CHIPY_HI, PHA, DELTPHA1, and} \\
\text{DELTPHA2.}
\end{array} \right. \quad (614)$$

$$\delta_2 = \left(\frac{\text{PHA} - \text{PHA}_m[r]}{\text{PHA}_{m+1}[r] - \text{PHA}_m[r]} \right) (\text{DELTPHA2}_{m+1}[r] - \text{DELTPHA2}_m[r]) + \quad (615)$$

$$\text{DELTPHA2}_m[r], \quad (616)$$

$$\epsilon = \text{is a uniform random deviate in the range } [0, 1), \quad (617)$$

$$\left\{ \begin{array}{l}
\text{If } \mathbf{rand_pha} = \text{no}, \text{ then } \epsilon = 0.
\end{array} \right. \quad (618)$$

(d) If
$$\text{PHA} \geq 32767, \quad (619)$$

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

22. CORN_PHA:

(a) If
$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (620)$$

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

23. 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

$$\text{CCD_ID} = \text{CCD_ID}_{\text{gain},i}, \quad (621)$$

$$\text{CHIPX_MIN}_{\text{gain},i} \leq \text{CHIPX} \leq \text{CHIPX_MAX}_{\text{gain},i}, \quad \text{and} \quad (622)$$

$$\text{CHIPY_MIN}_{\text{gain},i} \leq \text{nint}(\text{CHIPY_ADJ}) \leq \text{CHIPY_MAX}_{\text{gain},i}, \quad (623)$$

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

$$\text{PHA}_{\text{gain},i}[j] \leq (\text{PHA} + \Delta p) < \text{PHA}_{\text{gain},i}[j + 1], \quad (624)$$

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:

$$\text{ENERGY} = \left(\frac{\text{PHA} + \Delta p - \text{PHA}_{\text{gain},i}[j]}{\text{PHA}_{\text{gain},i}[j+1] - \text{PHA}_{\text{gain},i}[j]} \right) (\text{ENERGY}_{\text{gain},i}[j+1] - \text{ENERGY}_{\text{gain},i}[j]) + \text{ENERGY}_{\text{gain},i}[j], \quad (625)$$

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 $\text{ENERGY} = 0$.

24. PI:

(a) If

$$\text{calculate_pi} = \text{yes}, \quad (626)$$

then

i.

$$\text{PI} = \text{int} \left(\frac{\text{ENERGY}}{\text{pi_bin_width}} \right) + 1, \quad (627)$$

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

ii. If

$$\text{PI} < 1, \quad (628)$$

then $\text{PI} = 1$.

iii. If

$$\text{PI} > \text{pi_num_bins}, \quad (629)$$

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

(b) If

$$\text{calculate_pi} = \text{no} \quad (630)$$

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

25. `pix_adj`:

(a) centroid:

i. If

$$\text{pix_adj} = \text{centroid} \quad (631)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (632)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT} \text{ or} \quad (633)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS} \text{ or} \quad (634)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (635)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} - w'[0] + w'[2] - w'[3] + w'[5] - w'[6] + w'[8] \text{ and} \quad (636)$$

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} - w'[0] - w'[1] - w'[2] + w'[6] + w'[7] + w'[8], \quad (637)$$

where

$$w'[j] = \frac{w[j]}{\sum_{j=0}^8 w[j]}, \quad (638)$$

$$w[j] = \begin{cases} p[j] & \text{if the pixel is valid} \\ 0 & \text{if the pixel is invalid,} \end{cases} \quad (639)$$

$$p[j] = \begin{cases} \text{PHAS_ADJ}[j] & \text{if apply_cti} = \text{yes} \\ \text{PHAS}[j] & \text{if apply_cti} = \text{no,} \end{cases} \quad (640)$$

and the pixel is invalid if

$$\beta[j] = 0 \text{ or} \quad (641)$$

$$\text{STATUS}[0] = 1 \text{ or} \quad (642)$$

$$\text{STATUS}[1] = 1 \text{ or} \quad (643)$$

$$\text{STATUS}[2] = 1 \text{ or} \quad (644)$$

$$\text{STATUS}[3] = 1 \text{ or} \quad (645)$$

$$\text{STATUS}[4] = 1 \text{ or} \quad (646)$$

$$\text{STATUS}[11] = 1 \text{ or} \quad (647)$$

$$\text{STATUS}[13] = 1 \text{ or} \quad (648)$$

$$\text{STATUS}[14] = 1 \text{ or} \quad (649)$$

$$\text{STATUS}[15] = 1 \text{ or} \quad (650)$$

$$\text{STATUS}[16] = 1. \quad (651)$$

ii. If

$$\text{pix_adj} = \text{centroid and} \quad (652)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT}, \quad (653)$$

then

$$\text{TIME} = \text{TIME} + (w'[0] + w'[1] + w'[2] - w'[6] - w'[7] - w'[8]) \times \text{TIMEDEL}_{\text{in}}. \quad (654)$$

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

$$\text{pix_adj} = \text{edser} \quad (655)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (656)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED or} \quad (657)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (658)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (659)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (660)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (661)$$

then

A. If

$$\text{ENERGY} \neq \text{NaN and} \quad (662)$$

$$\text{ENERGY} > 0, \quad (663)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} + \left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta X[k+1] - \Delta X[k]) + \Delta X[k] \quad (664)$$

and

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k] \quad (665)$$

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

$$\text{ENERGY} \geq E[k] \text{ and} \quad (666)$$

$$\text{ENERGY} < E[k+1]. \quad (667)$$

Note that if

$$\text{ENERGY} \leq E[0], \quad (668)$$

then $k = 0$. Similarly, if

$$\text{ENERGY} \geq E[\text{NPOINTS}_{\text{subpix}} - 2], \quad (669)$$

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

B. If

$$\text{ENERGY} = \text{NaN or} \quad (670)$$

$$\text{ENERGY} \leq 0, \quad (671)$$

then the `CHIPX_ADJ` and `CHIPY_ADJ` coordinates are not modified.

ii. If

$$\text{pix_adj} = \text{edser} \quad (672)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (673)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (674)$$

then

A. If

$$\text{ENERGY} \neq \text{NaN and} \quad (675)$$

$$\text{ENERGY} > 0, \quad (676)$$

then

$$\text{TIME} = \text{TIME} - \left(\left(\frac{\text{ENERGY} - E[k]}{E[k+1] - E[k]} \right) (\Delta Y[k+1] - \Delta Y[k]) + \Delta Y[k] \right) \times \text{TIMEDEL} \quad (677)$$

B. If

$$\text{ENERGY} = \text{NaN or} \quad (678)$$

$$\text{ENERGY} \leq 0, \quad (679)$$

then the TIME is not modified.

(c) none:

If

$$\text{pix_adj} = \text{none}, \quad (680)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ and} \quad (681)$$

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ and} \quad (682)$$

$$\text{TIME} = \text{TIME.} \quad (683)$$

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

$$\text{pix_adj} = \text{randomize}, \quad (684)$$

then

$$\text{CHIPX_ADJ} = \text{CHIPX_ADJ} + \epsilon_x \text{ and} \quad (685)$$

$$\text{CHIPY_ADJ} = \text{CHIPY_ADJ} + \epsilon_y, \quad (686)$$

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

ii. If

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (687)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (688)$$

then

$$\text{TIME} = \text{TIME} - \epsilon_y \times \text{TIMEDEL}_{\text{in}}. \quad (689)$$

(e) If

$$\text{CHIPX_ADJ} < 0.5, \quad (690)$$

then

$$\text{CHIPX_ADJ} = 1. \quad (691)$$

(f) If

$$\text{CHIPX_ADJ} \geq 1024.5, \quad (692)$$

then

$$\text{CHIPX_ADJ} = 1024. \quad (693)$$

(g) If $\text{CHIPY_ADJ} < 0.5,$ (694)

then

$$\text{CHIPY_ADJ} = 1. \quad (695)$$

(h) If

$$\text{CHIPY_ADJ} \geq 1024.5, \quad (696)$$

then

$$\text{CHIPY_ADJ} = 1024. \quad (697)$$

26. X and Y:

(a) If

$$\text{stop} = \text{sky}, \quad (698)$$

then

i. If

$$\text{DATAMODE}_{\text{in}} = \text{FAINT or} \quad (699)$$

$$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS or} \quad (700)$$

$$\text{DATAMODE}_{\text{in}} = \text{GRADED or} \quad (701)$$

$$\text{DATAMODE}_{\text{in}} = \text{VFAINT}, \quad (702)$$

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

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT or} \quad (703)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (704)$$

then the values of X and Y are computed using the real-valued coordinates CHIPX_ADJ and $\text{CHIPY_TARG}_{\text{eff}}$ and the orientation of the telescope (i.e. RA, DEC, and ROLL) at the time TIME .

iii. If

$$\text{acaofffile} \neq \text{none} \quad (705)$$

and

$$\text{CONTENT}_{\text{in}} \neq \text{TGEVT1} \quad (706)$$

and

$$\text{TIME} < \text{TIME}_{\text{min}} \text{ or} \quad (707)$$

$$\text{TIME} \geq \text{TIME}_{\text{max}}, \quad (708)$$

then

$$\text{X} = \text{NaN and} \quad (709)$$

$$\text{Y} = \text{NaN}. \quad (710)$$

iv. If

$$\text{acaofffile} \neq \text{none} \quad (711)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ and} \quad (712)$$

$$\text{CHIPY_TG} = \text{NaN}, \quad (713)$$

then

$$X = \text{NaN} \text{ and} \quad (714)$$

$$Y = \text{NaN}. \quad (715)$$

27. SKY_1D:

(a) If

$$\text{stop} = \text{sky} \quad (716)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (717)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED}, \quad (718)$$

then

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

ii. If

$$\text{acaofffile} \neq \text{none} \quad (719)$$

and

$$\text{CONTENT}_{\text{in}} \neq \text{TGEVT1} \quad (720)$$

and

$$\text{TIME} < \text{TIME}_{\text{min}} \text{ or} \quad (721)$$

$$\text{TIME} \geq \text{TIME}_{\text{max}}, \quad (722)$$

then

$$\text{SKY_1D} = \text{NaN}. \quad (723)$$

iii. If

$$\text{acaofffile} \neq \text{none} \quad (724)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ and} \quad (725)$$

$$\text{CHIPY_TG} = \text{NaN}, \quad (726)$$

then

$$\text{SKY_1D} = \text{NaN}. \quad (727)$$

28. DETX and DETY:

(a) If

$$\text{stop} = \text{det or} \tag{728}$$

$$\text{stop} = \text{tan or} \tag{729}$$

$$\text{stop} = \text{sky,} \tag{730}$$

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

$$\text{acaofffile} \neq \text{none} \tag{731}$$

and

$$\text{CONTENT}_{\text{in}} \neq \text{TGEVT1} \tag{732}$$

and

$$\text{TIME} < \text{TIME}_{\text{min}} \text{ or} \tag{733}$$

$$\text{TIME} \geq \text{TIME}_{\text{max}}, \tag{734}$$

then

$$\text{DETX} = \text{NaN and} \tag{735}$$

$$\text{DETY} = \text{NaN.} \tag{736}$$

iii. If

$$\text{acaofffile} \neq \text{none} \tag{737}$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1 and} \tag{738}$$

$$\text{CHIPY}_{\text{TG}} = \text{NaN,} \tag{739}$$

then

$$\text{DETX} = \text{NaN and} \tag{740}$$

$$\text{DETY} = \text{NaN.} \tag{741}$$

29. TDETX and TDETY:

(a) If

$$\text{stop} = \text{tdet or} \tag{742}$$

$$\text{stop} = \text{det or} \tag{743}$$

$$\text{stop} = \text{tan or} \tag{744}$$

$$\text{stop} = \text{sky} \tag{745}$$

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

$$\text{acaofffile} \neq \text{none} \quad (746)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (747)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (748)$$

and

$$\text{CONTENT}_{\text{in}} \neq \text{TGEVT1} \quad (749)$$

and

$$\text{TIME} < \text{TIME}_{\text{min}} \text{ or} \quad (750)$$

$$\text{TIME} \geq \text{TIME}_{\text{max}}, \quad (751)$$

then

$$\text{TDETX} = \text{NULL} \text{ and} \quad (752)$$

$$\text{TDETY} = \text{NULL}. \quad (753)$$

iii. If

$$\text{acaofffile} \neq \text{none} \quad (754)$$

and

$$\text{DATAMODE}_{\text{in}} = \text{CC33_FAINT} \text{ or} \quad (755)$$

$$\text{DATAMODE}_{\text{in}} = \text{CC33_GRADED} \quad (756)$$

and

$$\text{CONTENT}_{\text{in}} = \text{TGEVT1} \text{ and} \quad (757)$$

$$\text{CHIPY_TG} = \text{NaN}, \quad (758)$$

then

$$\text{TDETX} = \text{NULL} \text{ and} \quad (759)$$

$$\text{TDETY} = \text{NULL}. \quad (760)$$

1.5.4 Write outfile

1. Unexpected CHIPX values:

If the counter associated with equations 168 and 169 is greater than zero, then `acis_process_events` produces a warning message that includes the number of events that have unexpected CHIPX values.

2. Unexpected CHIPY values:

If the counter associated with equations 196, 197, 199, 200, 201, 202, 205, and 206, is greater than zero, then `acis_process_events` produces a warning message that includes the number of events that have unexpected CHIPY values.

3. PIX_ADJ:

(a) If

pix_adj = centroid, (761)

then

PIX_ADJ = CENTROID. (762)

(b) If

pix_adj = edser, (763)

then

PIX_ADJ = EDSER. (764)

(c) If

pix_adj = none, (765)

then

PIX_ADJ = NONE. (766)

(d) If

pix_adj = randomize, (767)

then

PIX_ADJ = RANDOMIZE. (768)

4. RAND_SKY:

(a) If

pix_adj = centroid, (769)

then

RAND_SKY = 0.0. (770)

(b) If

pix_adj = edser, (771)

then

RAND_SKY = 0.0. (772)

(c) If

pix_adj = none, (773)

then

RAND_SKY = 0.0. (774)

(d) If $\text{pix_adj} = \text{randomize}$, (775)

then

$\text{RAND_SKY} = 0.5$. (776)

5. **TIME_ADJ**:

(a) Timed-exposure mode:
If

$\text{DATAMODE}_{\text{in}} = \text{FAINT}$ or (777)

$\text{DATAMODE}_{\text{in}} = \text{FAINT_BIAS}$ or (778)

$\text{DATAMODE}_{\text{in}} = \text{GRADED}$ or (779)

$\text{DATAMODE}_{\text{in}} = \text{VFAINT}$, (780)

then

$\text{TIME_ADJ} = \text{NONE}$. (781)

(b) Continuous-clocking mode:

i. Set

$\text{TIME_ADJ} = \text{MIDCHIP}$. (782)

ii. If

$\text{OBS_MODE}_{\text{in}} = \text{pointing}$ and (783)

$\text{acaofffile} \neq \text{none}$ (784)

then

$\text{TIME_ADJ} = \text{TARGET}$. (785)

iii. If

$\text{OBS_MODE}_{\text{in}} = \text{pointing}$ and (786)

$\text{CONTENT}_{\text{in}} = \text{TGEVT1}$, (787)

then

$\text{TIME_ADJ} = \text{GRATING}$. (788)

2 TBD

- Add the graded mode cti adjustment.
- Include all timed exposure mode processing.
- 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 β in PHA_RO the same as the β in PHA?
- Should something be done about SKY_1D?