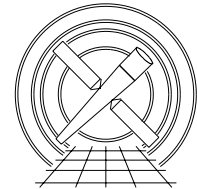




MIT Kavli Institute



Chandra X-Ray Center

MEMORANDUM

February 20, 2009

To: Jonathan McDowell, SDS Group Leader
From: Glenn E. Allen, SDS
Subject: acis_build_badpix
Revision: 1.6
URL: <http://space.mit.edu/CXC/docs/docs.html#abb>
File: /nfs/cxc/h2/gea/sds/docs/memos/memo_acis_build_badpix_1.6.tex

To make it easier to customize observation-specific bad-pixel files, the tool `acis_build_badpix` includes two enhancements. One enhancement permits users to ignore selected bad-pixel criteria. For example, a user can choose not to include cosmic-ray afterglows, hot pixels, and/or node boundaries in the output bad-pixel file. This functionality is embodied in the parameter `bitflag`. The same parameter can be used to prevent the code from identifying the neighbors of bad pixels as bad. For example, it is possible to prevent the code from flagging the columns next to bad columns as bad, but still flag pixels next to other bad pixels as bad.

A second enhancement enables users to include or exclude specific regions from the output file by preparing a supplemental file that includes a list of the regions of interest. The name of the supplemental input file is specified via the parameter `usrfile`.

1 Changes to `acis_build_badpix`

1.1 Additional Parameters

1. `usrfile,f,h`, “none”,,, “User-specified supplemental bad-pixel file (NONE — none — <filename>)”
2. `bitflag,s,h`, “0000000000000022221100020022222”,,, “A 32-character string where 0=exclude pixel, 1=include pixel, but not its neighbors and 2=include pixel and its neighbors”

1.2 Additional Input

The user has the option of specifying a supplemental input file which includes one row for each pixel, column or region. Each row has exactly nine columns that are tab or space delimited. The columns are, from left to right along a row,

1. `CCD_ID`,
2. `CHIPX_LO`,
3. `CHIPX_HI`,
4. `CHIPY_LO`,

5. CHIPY_HI,
6. TIME,
7. TIME_STOP,
8. BIT and
9. ACTION

1.3 Processing

1.3.1 `usrfile`

If the parameter `usrfile` is not “none” or “NONE” and specifies an existing file, then read each row of the file. If an error is detected in any row of the file, then write a warning message, ignore the entire file and continue processing. The possible error conditions include a row that

- does not have nine input values,
- has a `CCD_ID` that is not one of the `CCD_IDs` included in the list of active `CCDs` in the “PBK” extension of the parameter-block file,
- has a chip coordinate < 1 or > 1024 ,
- has a lower chip coordinate $>$ the corresponding upper chip coordinate,
- has a beginning or ending time < 0 ,
- has a beginning time \geq the ending time,
- has `BIT` < 0 or > 31 or
- has an `ACTION` other than “include” or “exclude.”

Note that the tool is not sensitive to the case of `ACTION`. If `ACTION` = “include,” then modify the output bad-pixel file to include a region with the following attributes.

- The input `CCD_ID` determines which extension is modified.
- The `SHAPE` is “point” if `CHIPX_LO` = `CHIPX_HI` and `CHIPY_LO` = `CHIPY_HI`. Otherwise, `SHAPE` = “rectangle.”
- The value of `COMPONENT` is determined in the usual manner.
- The two values in the double-valued vector column `CHIPX` are `CHIPX_LO` and `CHIPX_HI`.
- The two values in the double-valued vector column `CHIPY` are `CHIPY_LO` and `CHIPY_HI`.
- The value of `TIME` is the same as the input value unless the input values of `TIME` and `TIME_STOP` are both zero. In this case, set `TIME` equal to the time associated with the beginning of the observation.
- The value of `TIME_STOP` is the same as the input value unless the input values of `TIME` and `TIME_STOP` are both zero. In this case, set `TIME_STOP` equal to the time associated with the end of the observation.
- The specified `BIT` of `STATUS` is set to one. Note that other bits may be set to one for the region, if it is bad for other reasons.

If `ACTION` = “exclude,” then modify the output bad-pixel file so that any pixel in the region specified by `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO` and `CHIPY_HI` does not have the specified bit set to one for the time interval from `TIME` to `TIME_STOP`. (The output may become complicated if the time interval is only a subset of the observation.) Note that the rules about `TIME` = 0 and `TIME_STOP` = 0 also apply if `ACTION` = “exclude.”

If more than one row of the `usrfile` describes the same pixel, then apply the sequence of actions in order.

1.3.2 bitflag

Verify that the contents of the parameter `bitflag` are valid. This parameter should be a thirty-two character string, where the characters correspond to STATUS bits 0–31 from right to left. The acceptable and default values for each character are listed below.

Bit	Default value	Valid values	Comments
0–4	2	0–2	
5–6	0	0–1	A “2” doesn’t make sense.
7	2	1–2	Zero would be confusing if a <code>usrfile</code> is specified.
8 ^a	0	0	Adjacent pixels
9–10	0	0	Obsolete
11–12	1	0–2	
13–16	2	0–2	
17–31	0	0	Unused

^a The use of bit 8 of the parameter `bitflag` is not appropriate. Pixels adjacent to a bad pixel have STATUS bit 8 set if and only if the corresponding `bitflag` value for the pixel is “2.”

The action associated with each character is:

- 0: Ignore a bad pixel or column. Do not write such regions to the output file unless they satisfy another condition. If they do satisfy another condition, then the STATUS bit(s) for which `bitflag` = “0” are set to zero. For example, if the characters in the `bitflag` associated with STATUS bits 14 (hot pixels) and 15 (afterglows) are “0,” then exclude hot pixels and afterglows from the output bad-pixel file and do not set STATUS bits 14 and 15 to one for any pixels.
- 1: Include a bad pixel or column, but not the pixels or columns adjacent to it. For example, if the character in the `bitflag` associated with STATUS bit 1 (bad columns) is “1,” then the bad columns at `CHIPX` = 66, 496–8 and 738 for `CCD_ID` = 7 are marked as bad in the output file, but not the columns at `CHIPX` = 65, 67, 495, 499, 737 and 739. STATUS bit 8 is set to zero for the columns at `CHIPX` = 66, 496–8 and 738.
- 2: Include a bad pixel or column and the pixels or columns adjacent to it. For example, if the character in the `bitflag` associated with STATUS bit 1 (bad columns) is “2,” then the bad columns at `CHIPX` = 66, 496–8 and 738 for `CCD_ID` = 7 are marked as bad in the output file. In addition, the columns at `CHIPX` = 65–67, 495–499 and 737–739 have STATUS bit 8 set to one.

After the input in the `usrfile` and other files has been processed, set the adjacent bits using the standard rules, within the constraints of the parameter `bitflag`.