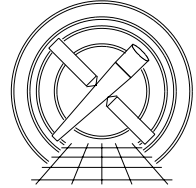




MIT Kavli Institute



Chandra X-Ray Center

## MEMORANDUM

October 6, 2009

**To:** Jonathan McDowell, SDS Group Leader  
**From:** Glenn E. Allen, SDS  
**Subject:** `acis_build_badpix`  
**Revision:** 2.5  
**URL:** <http://space.mit.edu/CXC/docs/docs.html#abb>  
**File:** `/nfs/cxc/h2/gea/sds/docs/memos/memo_acis_build_badpix_2.5.tex`

To make it easier to customize observation-specific bad-pixel files, the tool `acis_build_badpix` includes the new parameters `bitflag` and `usrfile`. The parameter `bitflag` enables users to ignore selected types of bad pixels. For example, a user can choose not to include cosmic-ray afterglows, hot pixels, and/or node boundaries in the output bad-pixel file. The same parameter can also be used to prevent `acis_build_badpix` from identifying the neighbors of bad pixels as bad. For example, it is possible to identify the pixels next to bad pixels as bad, but not the columns next to bad columns.

The parameter `usrfile` can be used to specify the name of an optional, user-created, input file. The input file contains a list of instructions that cause `acis_build_badpix` to add specific bad pixels to and/or remove specific bad pixels from the output bad-pixel file.

## 1 Changes to `acis_build_badpix`

### 1.1 Additional Parameters

1. `usrfile,f,h,"none",,,`,"An optional, user-created, bad-pixel file (NONE — none — <filename>)"
2. `bitflag,s,h,"000000000000000022221100020022222",,,`,"A 32-character string where 0=exclude pixel, 1= include pixel, but not its neighbors, and 2=include pixel and its neighbors"

Each character in the parameter `bitflag` corresponds to one of the 32 STATUS bits. These characters are associated with bits 0 to 31, from right to left. The acceptable and default values for each bit are listed in Table 1.

### 1.2 Additional Input

The parameter `usrfile` can be used to specify an optional, user-created, input file that has one or more rows. Each row has exactly nine space- or tab-delimited<sup>1</sup> items. From left to right, these items are

1. `CCD_ID`,

---

<sup>1</sup>The tool may accept other delimiters as well.

Table 1: Acceptable and default `bitflag` characters

Bit	Default value	Valid values	Comments
0-4	2	0-2	
5-6	0	0	These are in the <code>msk1.fits</code> file.
7	2	1-2	A “0” would be confusing with a <code>usrfile</code> .
8	0	0	Specified by using a “2” for the other bits
9-10	0	0	Obsolete
11-12	1	0-2	
13-16	2	0-2	
17-31	0	0	Unused

2. `CHIPX_LO`,
3. `CHIPX_HI`,
4. `CHIPY_LO`,
5. `CHIPY_HI`,
6. `TIME`,
7. `TIME_STOP`,
8. `BIT`, and
9. `ACTION`.

The integers `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, and `CHIPY_HI` specify the region of interest. `TIME` and `TIME_STOP` describe the time interval during which the specified region is affected. The `ACTION` string determines whether the specified `BIT` is set or unset for the region during this interval. Note that `acis_build_badpix` is not sensitive to the case of the `ACTION` string.

### 1.3 Processing

1. Verify that there are no input errors.
  - a. Errors associated with the parameter `usrfile` include rows that
    - do not have nine input items<sup>2</sup>,
    - have 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,
    - have `CHIPX_LO < 1`, `CHIPX_HI > 1024`, or `CHIPX_HI < CHIPX_LO`,
    - have `CHIPY_LO < 1`, `CHIPY_HI > 1024`, or `CHIPY_HI < CHIPY_LO`,
    - have a region, defined by `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, and `CHIPY_HI`, which overlaps another region in the `usrfile`,
    - have `TIME < 0` or `TIME_STOP ≤ TIME`<sup>3</sup>,
    - have `BIT < -1`<sup>4</sup>, `BIT = 5`, `BIT = 6`, `BIT = 8`, `BIT = 9`, `BIT = 10`, or `BIT > 16`,
    - have an `ACTION` other than “include” or “exclude,”

<sup>2</sup>If there are more than nine input items, then all items after the ninth are ignored.

<sup>3</sup>`TIME_STOP` may be equal to `TIME` if and only if `TIME = TIME_STOP = 0`.

<sup>4</sup>A row with `BIT = -1` is valid if and only if `ACTION = “exclude.”` If `ACTION = “include,”` then `BIT` cannot be less than zero.

- have ACTION = “include,” but the `bitflag` character for the specified bit is “0,”
- have BIT = 11 and (`CHIPX_LO` < 512 or `CHIPX_HI` > 513),
- have BIT = 12 and one or more of the values in the range from `CHIPX_LO` to `CHIPX_HI` is not equal to 256, 257, 768, or 769, and
- have BIT = 13 and `CHIPY_LO` < 513.

If there is an error with one or more rows of the `usrfile`, then exit with an error message.

b. Errors associated with the parameter `bitflag` include

- a string that does not have exactly 32 characters,
- a string where one or more of the characters has an invalid value. The valid values for each character are listed in Table 1, and
- a string in which every character is “0” except the character associated with STATUS bit 7 and the parameter `usrfile` = “none.”

If there is an error with the parameter `bitflag`, then exit with an error message.

2. As necessary, process the CALDB, bias, bias-parity error, and Level 0 event files. However, only set the appropriate STATUS bit to one if the corresponding character in `bitflag` is a “1” or “2.” At this stage in the processing, do not set STATUS bit 8 to one for pixels adjacent to bad pixels.
3. If necessary, process each row of the `usrfile`.
  - a. If a row has ACTION = “include,” then the specified STATUS bit is set to one for the pixels in the region defined by `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, and `CHIPY_HI` for the time interval from `TIME` to `TIME_STOP`<sup>5</sup>. Note that the bit is set to one if and only if the corresponding character in `bitflag` is a “1” or “2.” If the `bitflag` character is “0,” then this represents an error in the `usrfile` (see step 1). Note that other STATUS bits may be set to one for pixels in the region. Again, do not set STATUS bit 8 to one for pixels adjacent to bad pixels at this stage.
  - b. If a row has ACTION = “exclude,” then set the specified STATUS bit to zero<sup>6</sup> for the pixels in the region defined by `CCD_ID`, `CHIPX_LO`, `CHIPX_HI`, `CHIPY_LO`, and `CHIPY_HI` for the time interval from `TIME` to `TIME_STOP`. It does not matter if the corresponding `bitflag` character is a “0,” “1,” or “2.” Note that a pixel may remain bad if other STATUS bits are set to one. The output may become complicated if the specified time interval is only a subset of the observation.
4. Once all of the input files have been processed, set STATUS bit 8 to one for all pixels that are adjacent to bad pixels provided that the corresponding value of `bitflag` is “2.” If a bad pixel has more than one STATUS bit set to one and the `bitflag` character for at least one of the STATUS bits is “2,” then set STATUS bit 8 to one for the adjacent pixels.

---

<sup>5</sup>If `TIME` = `TIME_STOP` = 0, then `TIME` and `TIME_STOP` are the times associated with the beginning and end of the observation, respectively.

<sup>6</sup>If BIT = -1, then set all 32 STATUS bit to zero.