VIS Flags Documentation

Last update: 2024-04-10, maintainer: mottet@iap.fr

Important note

This page may not be up-to-date. Please refer to the flags description in the DPDD: http://st-dm.pages.euclid-sgs.uk/data-product-doc/visdpd/dpcards/vis_flagmap.html. Please note that the DPDD may not be up-to-date either :-) , the ultimate and always up-to-date source of information about flagmaps is the source code: VIS_ImageTools_M/FlagMap.py gitlab

Related pages

- SGS ImageBitMasks
- VIS_ImageTools_M/FlagMap.py gitlab
- issue #13589: Common flags definition in the DM
- task #5244: Bit mask propagation during data resampling and stacking

Flag bitmask values

Bit	Bit Mask	Flag Value	Flag Name	Invalid	Description
0	0x00000001	1	INVALID		automatic interface flag identifying pixels with an arbitrary value which shall not be used for anything
1	0x00000002	2	HOT	X	hot pixel: defective always bright pixel
2	0x00000004	4	COLD	X	cold pixel: defective unresponsive pixel
3	0x00000008	8	SAT	X	saturated pixel: pixel near or above blooming threshold
4	0x00000010	16	COSMIC	X	cosmic ray: pixel affected by a cosmic ray
5	0x00000020	32	GHOST	X	optical ghost: pixel affected by an optical ghost
6	0x00000040	64	QUADEDGE		quadrant edge: pixel near a quadrant edge, which causes pixel-size and nonlinear response variations
7	0x00000080	128	BAD_COLUMN	X	part of a column of bad pixels which have an abnormal behaviour, e.g. hot, cold, low noise, high noise
8	0x00000100	256	BAD_CLUSTER	X	secondary bad pixel which is part of a cluster of immediate

					neighbours of HOT or COLD pixels
9	0x00000200	512	CR_REGION	X	pixel belonging to a region in which detected cosmic rays per pixel ratio is too high for correct processing
12	0x00001000	4,096	OVRCOL	X	overflow column: pixel affected by a neighbouring saturated pixel
15	0x00008000	32,768	CHARINJ	X	charge injection: pixel part of charge injection line, not exposed
17	0x00020000	131,072	SATXTALKGHOST	X	saturated crosstalk ghost: pixel affected by electronic crosstalk caused by a saturated pixel
18	0x00040000	262,144	STARSIGNAL		star signal: pixel affected by a star PSF
21	0x00200000	2,097,152	ADCMAX	X	analog-to-digital-converter maximum: pixel at maximum ADC value (65,535) before VIS processing
22	0x00400000	4,194,304	NO_DATA	X	no data available for this pixel (being 0 ADU in VIS input raw frame), due to transmission error, truncated/incomplete RAW file, failing quadrant, CCD or ROE, or anything else
23	0x00800000	8,388,608	STITCHBLOCK		Photolithographic stitch-block boundary: pixel may have a non-nominal size
24	0x01000000	16,777,216	OBJECTS		Detected objects in science image, currently non-zero pixels in OBJECTS SExtractor check-image

Detailed flags description

bit 0 - INVALID

Convenience flag, combination of HOT, COLD, SAT, COSMIC, GHOST, BAD_COLUMN, BAD_CLUSTER, CR_REGION, OVRCOL, CHARINJ, SATXTALKGHOST, ADCMAX, NO_DATA. **Pixels flagged wih INVALID have an arbitrary value and shall not be used for anything. ANY AND ALL ODD NUMBER FLAG VALUES INDICATE AN INVALID PIXEL**

bit 1 - HOT

Hot pixels are pixels which are much brighter than the average of the quadrant pixels. They are detected in dark frames and in flat exposures. In the master flat frame, HOT pixels are pixels which value is over 1.15. In the master dark frame, HOT pixels are pixels above 0.01 e-/s (#22191-66) or all the pixels in a column containing 200 or more HOT pixels (#22191-69).

bit 2 - COLD

Cold pixels are under-responsive pixels. Like hot pixels, they are detected in the master flat product, as having a value lower than 0.85.

bit 3 - SAT

The saturated pixel detection is the first operation of the VIS PF. The **SAT** flag is set for pixels which have a value greater than the blooming threshold measured by the VIS_BloomingCalibration pipeline.
The blooming threshold value is copied in the FITS file HDU header of the corresponding quadrant:

```
SATLEVEL = 192000.0  / pixel saturation level in e-
SATURATE = 61935     / pixel saturation level in ADU
```

The **SATURATE** FITS keyword is used by SExtractor to filter saturated objects.

bit 4 - COSMIC

Pixel affected by a cosmic ray as detected by LA Cosmicx (https://github.com/cmccully/lacosmicx). X-rays and proton showers are flagged by the same **COSMIC** value.

bit 5 - GHOST

The **GHOST** flag identifies pixels affected by an optical ghost image produced by the defocused reflection of a very bright object on the back face of the dichroic plate. The algorithm description and illustrative plots and results can be found here

bit 6 - QUADEDGE

Pixel near a quadrant edge, which causes pixel-size and nonlinear response variations, see redmine issues #23824 and #9826 for more details.

bit 7 - BAD_COLUMN

Part of a column of bad pixels which have an abnormal behaviour, e.g. hot, cold, low noise, high noise (#24926-19).

bit 8 - BAD_CLUSTER

Secondary bad pixel which is part of a cluster of immediate neighbours of HOT or COLD pixels (#24926-19)

bit 9 - CR_REGION

Pixel belonging to a region in which detected cosmic rays per pixel ratio is too high for correct processing (#24332). Currently, the CR per pixel ratio threshold is 3%. This will change in the future, as CR flagging evolves.

bit 12 - OVRCOL

Overflow column: pixel affected by an adjacent saturated pixel in the same column, flag set by VIS_SatPixels.

bit 15 - CHARINJ

Flags pixels belonging to charge injection lines. Charge injection lines are 2 lines of pixels per quadrant which are at the opposite side of the readout register. They can't be read (see below) but are physically present in the CCD images, in the form of 4 consecutive horizontal lines at the center of the CCD images (2 quadrants glued head-to-head, see EUCL-EC-ICD-8-001 Data Product Description v0.1 documentation).

The charge injection lines can be set to static voltages through the configuration of the voltages IG1 and IG2. They can dump charge on the image area pixels if the timings of the injection and the voltages of IG1, IG2 and Injection Drain are set as needed for that to happen, but you can't actually read them out because they don't have the phase electrodes which make possible to clock charge, as

the image area pixels have (the "Iphi's") (Ruyman Azzollini 14 Dec. 2020).

bit 17 - SATXTALKGHOST

Flags a pixel affected by electronic crosstalk from a saturated source pixel in a different quadrant of the same ROE. Since electronic crosstalk happens before Analog-to-Digital Conversion, if a source pixel is ADC saturated (65,535), we can't know its 'real' value when it affects the victim pixel. So this flag identifies pixels affected by non-negligible electronic crosstalk at a level that can't be estimated and corrected for.

bit 18 - STARSIGNAL

Pixels affected by a PSF from all stars in the input GAIA catalogue are flagged using a 1024x1024 broadband PSF image scaled in brightness using the magnitude of the star and then applying a 2 ADU threshold. Note that this flagging misses long diffraction star spikes.

bit 21 - ADCMAX

Set when a pixel value is 65,535 ADU (maximum 16 bit Analog-to-Digital Converter value) in the VIS PF input raw frame. Please note that after VIS processing, some pixels can have a vlaue above 65,535 ADU without being saturated.

bit 22 - NO_DATA

This flag bit is set when a pixel in VIS input raw frame contains 0. This can indicate a transmission error either inside the satellite or between the satellite and the ground (lost packet), or a malfunctionning or switched off quadrant, CCD or ROE, or a prematurely closed or truncated data file on-board, or any missing data for any other reason.

bit 23 - STITCHBLOCK

Photolithographic stitch-block boundary, in which pixels may have a different size, which may slightly affect locally the image astrometry and photometry. It's here for now just for documentation purposes, as it's not used anywhere in the VIS PF yet. Be warned, however, that you may not want to use these pixels to do very precise science.

bit 24 - OBJECTS

Objects detection map produced by SExtractor. It may contain any signal which can not be identified as sky background (eg. stars, galaxies, cosmic rays, optical ghosts, electronic crosstalk, hot and cold pixels, straylight, CTI, etc.). It's also produced by the VIS_FlagObjects pipeline running on short duration exposures, to flag sky signal in flat field images taken with the same pointing as the VIS_FlagObjects input exposure.

Software interface

VIS Flags bitmasks are defined in the VIS_ImageTools project, starting with version **3.7** (January 2021).

A standardisation across different MER input was decided in October 2021 (#13589#note-48) which led to the introduction of the INVALID flag, which broke flagmap compatibility between VIS PF v4.x and VIS PF 5.x.

They are declared in Python as a dictionary of 32 bit integer values. These integers are not unsigned because we don't need bit 31 yet, and unsigned int is not a native FITS datatype, it must be handled with a special BZERO card scaling (https://heasarc.gsfc.nasa.gov/docs/software/fitsio/c/c_user/node26.html), which might introduce handling issues.

```
VIS_FLAGS = {
# declaration automatically generated on Thu Mar 14 10:30:01 2024 by VIS_ImageTools_M/FlagMap.gene
rate_VIS_FLAGS_declaration()
# flag name                      bit     hexadecimal  decimal      invalid
  'GOOD':          np.int32( 0      ), # 0x00000000 |          0 |
  'INVALID':       np.int32( 1 <<  0), # 0x00000001 |          1 |
  'HOT':           np.int32( 1 <<  1), # 0x00000002 |          2 | X
  'COLD':          np.int32( 1 <<  2), # 0x00000004 |          4 | X
  'SAT':           np.int32( 1 <<  3), # 0x00000008 |          8 | X
  'COSMIC':        np.int32( 1 <<  4), # 0x00000010 |         16 | X
  'GHOST':         np.int32( 1 <<  5), # 0x00000020 |         32 | X
  'QUADEDGE':      np.int32( 1 <<  6), # 0x00000040 |         64 |
  'BAD_COLUMN':    np.int32( 1 <<  7), # 0x00000080 |        128 | X
  'BAD_CLUSTER':   np.int32( 1 <<  8), # 0x00000100 |        256 | X
  'CR_REGION':     np.int32( 1 <<  9), # 0x00000200 |        512 | X
```

```
  'OVRCOL':        np.int32( 1 << 12), # 0x00001000 |      4,096 | X
  'CHARINJ':       np.int32( 1 << 15), # 0x00008000 |     32,768 | X
  'SATXTALKGHOST': np.int32( 1 << 17), # 0x00020000 |    131,072 | X
  'STARSIGNAL':    np.int32( 1 << 18), # 0x00040000 |    262,144 |
  'ADCMAX':        np.int32( 1 << 21), # 0x00200000 |  2,097,152 | X
  'NO_DATA':       np.int32( 1 << 22), # 0x00400000 |  4,194,304 | X
  'STITCHBLOCK':   np.int32( 1 << 23), # 0x00800000 |  8,388,608 |
  'OBJECTS':       np.int32( 1 << 24), # 0x01000000 | 16,777,216 |
}
```

The new INVALID flag introduced in version 5.0 of the VIS PF is a combination of other flags, which indicates that a pixel must not be used for science or image stacking. The list of flag names to combine to produce the INVALID flag is declared in the same FlagMap.py python file, along with tools to use it:

```
# names of flags used to build the INVALID flag
INVALID_BITNAMES = ( 'HOT', 'COLD', 'SAT', 'COSMIC', 'GHOST', 'BAD_COLUMN', 'BAD_CLUSTER',
                     'CR_REGION', 'OVRCOL', 'CHARINJ', 'SATXTALKGHOST', 'ADCMAX', 'NO_DATA')

# INVALID flag bitmask is a bitwise OR of all flags in INVALID_BITNAMES
INVALID_BITMASK = np.int32( np.bitwise_or.reduce( [VIS_FLAGS[bitname] for bitname in
INVALID_BITNAMES]))

 build/update the INVALID flag in 'flagmap' by combining the flags in 'INVALID_BITNAMES'
def set_invalid_flag( flagmap):
  # clear old INVALID flag
  flagmap &= ~VIS_FLAGS["INVALID"]
  # set new flag
  flagmap[np.nonzero( flagmap & INVALID_BITMASK)] |= VIS_FLAGS["INVALID"]
  return

# set to 0 all the 'INVALID' flagged pixels in an array 'weighmap' with same shape as 'flagmap'
def set_invalid_pixels_to_zero( weightmap, flagmap):
  set_invalid_flag( flagmap)
  weightmap[ np.nonzero( flagmap & VIS_FLAGS["INVALID"])] = 0
  return
```

You can use the flag dictionary from any Python source code as follows:

```
from VIS_ImageTools_M.FlagMap import VIS_FLAGS

import numpy as np

for flag_name, value in VIS_FLAGS.items():
  if flag_name != "GOOD":
    print(f"| {int(np.log2(value)):2d} | 0x{value:08x} | {value:10,d} | {flag_name:14s} |")

print("STARSIGNAL" in VIS_FLAGS)

print(f"{VIS_FLAGS['ADCMAX']:_b}")

print(VIS_FLAGS["GHOST"].dtype)
```

gives:

```
|  0 | 0x00000001 |          1 | INVALID        |
|  1 | 0x00000002 |          2 | HOT            |
|  2 | 0x00000004 |          4 | COLD           |
|  3 | 0x00000008 |          8 | SAT            |
|  4 | 0x00000010 |         16 | COSMIC         |
|  5 | 0x00000020 |         32 | GHOST          |
|  6 | 0x00000040 |         64 | QUADEDGE       |
|  7 | 0x00000080 |        128 | BAD_COLUMN     |
|  8 | 0x00000100 |        256 | BAD_CLUSTER    |
|  9 | 0x00000200 |        512 | CR_REGION      |
| 12 | 0x00001000 |      4,096 | OVRCOL         |
| 15 | 0x00008000 |     32,768 | CHARINJ        |
| 17 | 0x00020000 |    131,072 | SATXTALKGHOST  |
```

18	0x00040000	262,144	STARSIGNAL
19	0x00080000	524,288	SATURATEDSTAR
20	0x00100000	1,048,576	CTICORRECTION
21	0x00200000	2,097,152	ADCMAX
22	0x00400000	4,194,304	NO_DATA
23	0x00800000	8,388,608	STITCHBLOCK
24	0x01000000	16,777,216	OBJECTS

```
True
10_0000_0000_0000_0000_0000
int32
```