

# epchain

June 2, 2019

## Abstract

The task **epchain** generates the EPIC-PN event list product making use of the tasks **atthkgen**, **epframes**, **badpixfind**, **badpix**, **epreject**, **epnoise**, **epevents**, **attcalc**, **evlistcomb**, **epfast**, **epexposure**, **epatplot**, **tabgtigen**, and **evselect**.

## 1 Instruments/Modes

Instrument	Mode
EPIC PN	all (IMAGING, TIMING, BURST)

## 2 Use

pipeline processing	no
interactive analysis	yes

## 3 Description

### 3.1 General

The **epchain** task chains and loops over all first-level EPIC PN tasks to produce an event list ready to be exported as a PPS product. The **epchain** script is executed with a number of command line input parameters (arguments to the script). All parameters are optional, the parameter order is arbitrary. Parameters are given in the form “parameter=value”. The implemented list of parameters and the corresponding default values are shown below. Note that in the case of the same parameter appearing twice on the command line the first instance is used (this behavior is different from other SAS tasks where the second instance is used).

Input files corresponding to the specified readout mode, CCD and exposure number are searched for in the given ODF directory. If desired **epchain** can also be directed to select all exposures or all exposures using the selected readout mode for processing. Badpixfind files (created by **badpixfind** and to be used by **badpix**) will be placed in **badpixset**. Output files are created in the current working directory.

The PN mini pipeline is sketched out in Fig. 1. At the beginning of the **epchain** script the task **atthkgen** creates an attitude history file, called **atthk.dat**, which will be used by **attcalc**. The main subchain

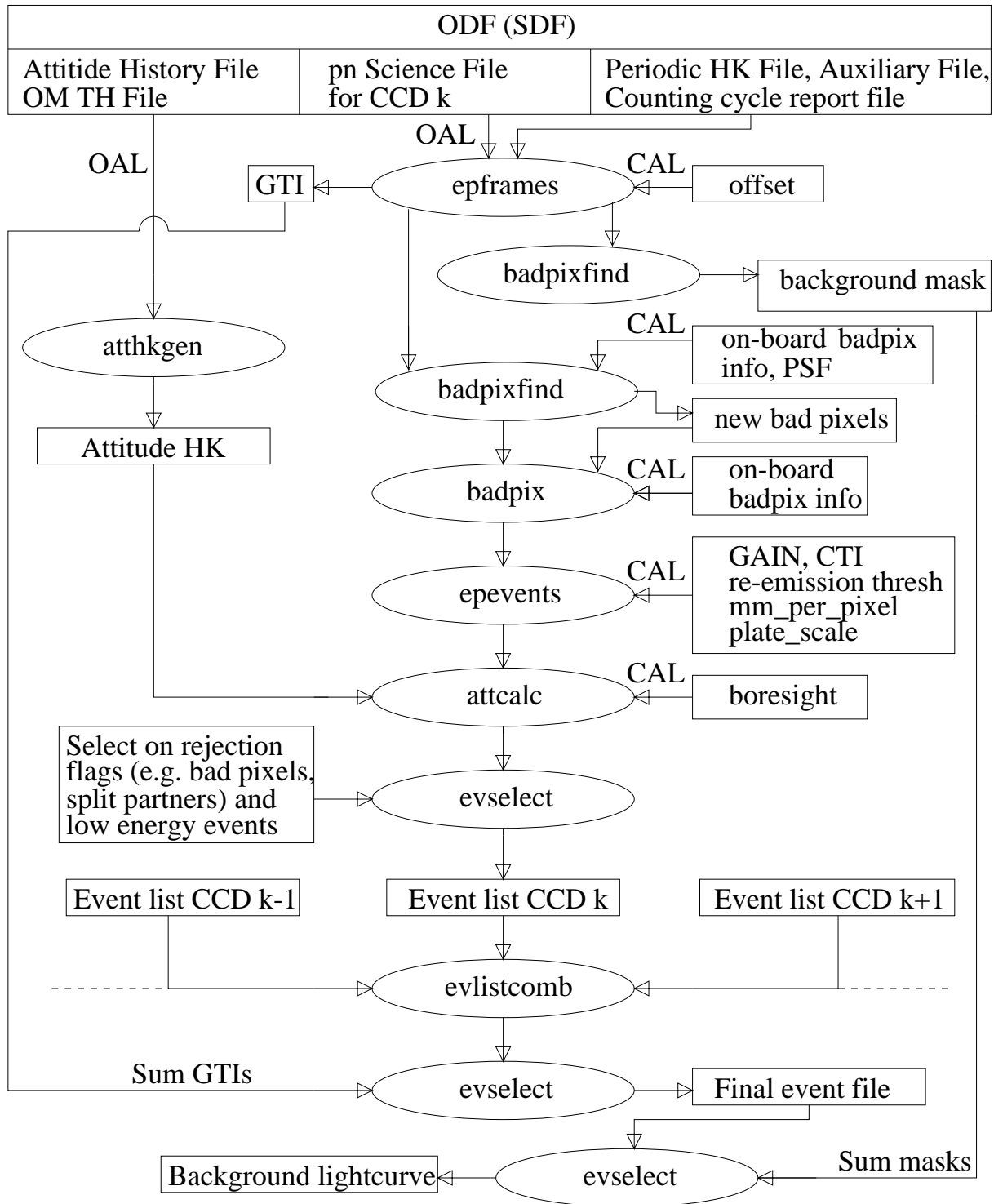
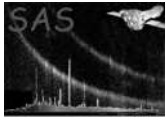
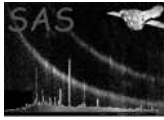


Figure 1: Organization of the EPIC-pn chain (only main tasks shown for clarity)



(**epframes**, **badpixfind**, **badpix**, **epevents** and **attcalc**) creates one event list for a single exposure and for a given list of CCDs from all the relevant ODF material and bad pixel lists.

The main subchain loops over all specified CCDs, calling in sequence:

1. **epframes** to process a CCD, exposure and datamode specific ODF file, creating the output events list `rawevents##.dat` and the GTI data set `gti##.dat` (`##` is the loop index, i.e. the CCD number currently in use)
2. **badpixfind** to find new bad pixel [optional]
3. **badpix** to process the rawevents list, adding the BADPIX extension
4. **epreject** to optionally correct the energy scale in specific pixels, and to flag soft flare events (turned off by default; see description of task **epreject** for details)
5. **epnoise** to optionally flag low energy detector noise (turned off by default; see description of task **epnoise** for details)
6. **epevents** to process the event list file, flag trailing events, perform pattern recognition, gain and CTI corrections, and compute linearized detector coordinates to create the calibrated events list `events##.dat`
7. **attcalc** to calculate the X and Y sky coordinates
8. **epfast** for RDCTI correction in FAST modes
9. **epexposure** for TIME randomization and EXPOSURE extension screening

Finally, making use of the common PN and MOS task **evlistcomb**, the CCD specific data sets are merged into a single events list. **evselect** selects all those events arriving in good time intervals and writes an output file according to XMM-SOC-ICD-0006-SSC (v2.1) ([1]).

It is highly recommended to also read the documentation of **epframes** and **epevents**.

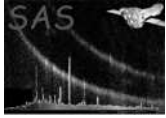
### 3.2 Out-of-time events

Depending on the observation mode a certain fraction of the events recorded are due to out-of-time events, i.e. are registered during readout of a CCD and thus will be assigned an incorrect position in RAWY and consequently an incorrect CTI correction. Typical values are 6.3% for full frame mode, and 2.3% for extended full frame mode (with frame time parameter 3).

If `withoutoftime=Y` the task **epevents** does not create the “normal” events file but uses the output of **epframes** instead to create an event file where all events are treated as out-of-time events. After the pattern recognition for the same TIME, PHA, and RAWX a new RAWY is simulated by randomly shifting the pattern along the RAWY axis and performing the gain and CTI correction afterwards.

This out-of-time events file has the same temporal variations and pattern distribution as the “normal” event file. All events have an energy correction corresponding to the (randomly distributed) RAWY column. You can apply a selection on both the “normal” and “out-of-time” event file, scale with a mode-dependent factor and subtract the “out-of-time” contribution from the other. You can create “out-of-time”-corrected spectra and images for an arbitrary time and energy range.

**Note:** When using **epchain** with parameter `withoutoftime=Y` it is highly recommended to follow the order as illustrated in the examples section (see Sect.3.7): first with parameter setting



`withoutoftime=Y keepintermediate=raw`  
and then second with  
`withoutoftime=N` (is default and can thus be omitted)  
making use of intermediate files.

The current (experimental) implementation does not automatically detect the instrument and bad pixel setting. The `RAWY` range to distribute the out-of-time events is instead derived from the `RAWY` range of the actual events in the data. The mode-dependent conversion factor has thus to be applied manually and is about 95% of the out-of-time event fraction (e.g. 0.060 for full frame mode) due to setting the top 10 rows to “bad”. This will be improved in later versions.

A different approach is done via the setting `withctisrcpos=Y`: all event energy corrections do not use the actual `RAWY` coordinate but the corresponding `RAWY` location of the source (`SRCPOS`) as determined by the task `epframes` (either automatically or via setting `withsrccoords=Y`). So all events are assumed to originate from the source position. Events that really occur at a `RAWY` different from `SRCPOS` get a wrong (usually too high) CTI correction instead. (Note, that this procedure is the correct handling for `TIMING` and `BURST` mode – where it is performed automatically). A spatial selection should be applied in the final event analysis to deal only with detector portions dominated by Out-of-Time events. `epchain` may be used with the setting `ccds=4 withctisrcpos=Y` to process only the CCD 4 where usually the bright source is located that causes the Out-of-Time events.

### 3.3 Incomplete ODFs

In the early phase of the XMM mission correct ODFs were not available. To be able to analyse EPIC-pn flight data obtained during the commissioning phase a couple of temporary parameters were introduced to mimic the interface to the housekeeping files (PMH, PAH) needed in `epframes` (see the corresponding task description for more details). All these parameters are *not relevant for data from the Guest Observer programme*. To reduce the number of warnings due to missing files and columns use the setting “`odfok=N hkok=N`”.

### 3.4 Creation of background spectrum and lightcurve; screening of periods of high background

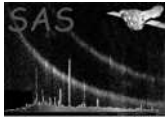
If requested (`runbackground=Y`), the task creates a background lightcurve, as well as a background spectrum for the imaging modes (FF, eFF, LW, SW) using background masks. These are not used for the fast modes as there is no useful background region, in which case the total lightcurve is produced. The usage of background masks can be controlled by parameter `withmask` (default value: yes). The background masks are created by task `badpixfind` using threshold parameters, appropriate for the detection of real X-ray sources. The background lightcurves and corresponding masks are created in the 7.0 - 15.0 keV energy band, while masks created in the 0.2-10.0 keV energy band are used for the creation of the background spectrum.

Depending on the individual observation and on the requested `timebinsize` one can then create a GTI file via `tabgtigen` (e.g.):

```
tabgtigen table=rate_bkg.fits gtiset=bkg_GTI.fits expression='COUNTS<500'
```

or using the count rate column

```
tabgtigen table=rate_bkg.fits gtiset=bkg_GTI.fits expression='RATE<8'
```



where the output rate is in counts  $\text{ks}^{-1} \text{ arcmin}^{-2}$ . Typical low-background rates for PN are of the order 3 - 5 counts  $\text{ks}^{-1} \text{ arcmin}^{-2}$  in the 7.0 to 15.0 keV energy range.

The GTI file can then be used to screen the event list via **evselect** (e.g.):

```
evselect expression='GTI(bkg_GTI.fits,TIME)' ...
```

As the GTIs strongly depend on the intended science this screening is not performed and only background masks, the lightcurve, and a GTI file are created by **epchain**.

### 3.5 Optical loading

As the final event file does not contain events below 150 eV, the information to assess the presence of optical loading in a straight forward way is not available in the event file anymore. Parameter **optloadingimage** therefore controls the optional creation of an image in the ADU 20–30 range where optical loading is easily visible.

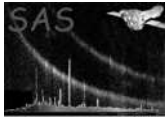
### 3.6 Missing offset maps in ODF or SDF

Sometimes offset maps are not available within the ODF/SDF – they may be been dumped instead in the observation before or after the one to be processed, with different ObsID. A work-around is implemented for **epreject** via an ASCII list of corresponding offset maps outside the ODF/SDF.

This is controlled via the parameter **withoffsetlist** (default “N”) that takes precedence over parameter **withoffsetmap** if set to “Y”: it opens the file specified in parameter **odilist** and reads the ODI name to be used from there. The input list is an ASCII with 2-digit CCD number followed by a blank and then the path to the offset map for that CCD (there can be more than one offset map in that file so that the input file can be used for the whole observation processing in **epchain**; only the first entry for a particular CCD is used; the order does not matter).

An example input file (you may call it 2462\_9246200004\_PNS00300ODI.ASC) useful for slew 9246200004 is listed below:

```
01 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40101ODI.FIT
02 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40102ODI.FIT
03 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40103ODI.FIT
04 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40104ODI.FIT
05 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40105ODI.FIT
06 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40106ODI.FIT
07 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40107ODI.FIT
08 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40108ODI.FIT
09 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40109ODI.FIT
10 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40110ODI.FIT
11 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40111ODI.FIT
12 /xmm/public/data/2462/0723780401/2462_0723780401_PNU40112ODI.FIT
```



### 3.7 Examples

Let us assume we have an observation 0084\_0099280101 with three exposures, one TIMING mode exposure PNS008, and two imaging exposures PNS010 (in Small Window Mode) and PNS018 (in Full Frame Mode). Here follow some typical calls to `epchain`:

```
epchain
```

This processes the first IMAGING mode exposure (i.e. PNS010) and is equivalent to one of the calls

```
epchain datamode=IMAGING odfaccess=odf exposure=010 schedule=S
epchain datamode=IMAGING odfaccess=oal exposure=1
epchain datamode=IMAGING exposure=1
epchain odfaccess=name odf=0084\_0099280101\_PNS01000AUX.FIT
```

If you are interested in the second IMAGING exposure then you may enter

```
epchain exposure=2
```

or (if you like it more verbose) any one of these:

```
epchain datamode=IMAGING odfaccess=odf exposure=018 schedule=S
epchain datamode=IMAGING odfaccess=oal exposure=2
epchain odfaccess=name odf=0084_0099280101_PNS01800AUX.FIT
```

Finally, the TIMING mode exposure (if existent) is processed via any of the calls

```
epchain datamode=TIMING
epchain datamode=TIMING exposure=1
epchain datamode=TIMING odfaccess=oal exposure=1
epchain datamode=TIMING odfaccess=odf exposure=008
epchain datamode=TIMING odfaccess=odf exposure=008 schedule=Y
epchain odfaccess=name odf=0084_0099280101_PNS00800AUX.FIT
```

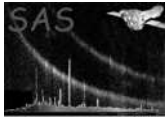
The commands

```
epchain odfaccess=all
epchain datamode=IMAGING odfaccess=all
```

will process all IMAGING mode exposures in the selected ODF in one call to `epchain`. All TIMING mode or BURST mode exposures (if available) may be selected accordingly. The call

```
epchain datamode=ALL odfaccess=all
```

will process all exposures of an ODF, independent of observing mode. Sometimes one is interested just in particular (e.g. central) CCDs:



```
epchain ccds=1,4,7,10 exposure=2 ...
```

or one does not want to search for bad pixels (e.g. for short exposures) and no background lightcurve:

```
epchain runbadpix=N getnewbadpix=N runbackground=N ...
```

To create the output filenames according to the ODF (i.e. P0099280101PNS010\*.FIT) use the parameter switch (this is the default in the meanwhile)

```
epchain usecanonicalnames=Y ...
```

Finally, the next example shows how to create a “normal” events file together with an “out-of-time” events file by re-using intermediate files (the order of the two calls to **epchain** is relevant):

```
epchain runbackground=N keepintermediate=raw withoutoftime=Y
epchain runatthkgen=N runepframes=N runbadpixfind=N runbadpix=N
```

Some special settings that may be useful for particular observations:

If you have a very soft source and you want to have also the very softest photons to be included in the final event list (e.g. lower threshold 100 eV instead of 150 eV) then (be prepared for a very large event file):

```
epchain screenlowthresh=0 ... # default: 150
```

If you have an offset pointing in TIMING or BURST mode (either intentionally or because the sky position of the source was not accurately known at the time of the scheduling e.g. for an “Target of Opportunity”) then the correct source position can be entered via parameters (see also Sect. 4) to ensure correct timing and CTI corrections:

```
epchain ... withsrccords=Y srcra=83.633208 srcdec=22.0124194
```

If you are interested in diagnostic images and plots and the EPEA rejection setup

```
epchain ... witheventmap=Y withphotonmap=Y withpatplot=Y showpah=Y
```

## 4 Parameters

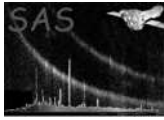
This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

### General

<b>odf</b>	no	string	\$SAS_ODF	
------------	----	--------	-----------	--

input ODF directory name



<b>odfaccess</b>	no	string	oal	oal/odf/name/all
------------------	----	--------	-----	------------------

how **oal** should access the ODF

<b>schedule</b>	no	string	S	S/U
-----------------	----	--------	---	-----

exposure schedule flag, only used if **odfaccess**=**odf**

<b>datamode</b>	no	string	IMAGING	IMAGING, TIMING, BURST or ALL
-----------------	----	--------	---------	-------------------------------

data modes to be processed

<b>exposure</b>	no	integer	1	
-----------------	----	---------	---	--

exposure number to be processed, see **oal** documentation

<b>ccds</b>	no	string	1-12	1-12
-------------	----	--------	------	------

list of CCDs to be processed, in a form like: 1,6-9,12 or 7,3,11

<b>keepintermediate</b>	no	string	none	all/raw/cal/clean/ notmerged/none
-------------------------	----	--------	------	--------------------------------------

indicator which type of intermediate files should not be deleted, default is to delete all intermediate files during or at the end of the processing

<b>withdefaultcal</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----

whether mode-dependent calibration-related settings should be used automatically, this will use for:  
 BURST: **withrdpha**="N", **withxrlcorrection**="Y", **runereject**="Y", **runepfast**="Y"  
 TIMING: **withrdpha**="Y", **withxrlcorrection**="Y", **runereject**="Y", **runepfast**="N"  
 IMAGING: **withrdpha**="N", **withxrlcorrection**="N", **runereject** as user-supplied, **runepfast**="N"

<b>usecanonicalnames</b>	no	boolean	Y	Y/N
--------------------------	----	---------	---	-----

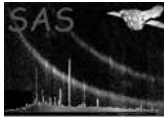
auto-create output filenames ?

<b>outset</b>	no	string	events.fits	
---------------	----	--------	-------------	--

name of output events file if **usecanonicalnames**=N

<b>ingtiset</b>	no	string		
-----------------	----	--------	--	--

name of input HK GTI file [not active yet]



<b>optloadingimage</b>	no	boolean	N	Y/N
------------------------	----	---------	---	-----

creation of diagnostic ADU 20-30 sky image to assess optical loading

<b>optloadingimageset</b>	no	string	optloading.img	
---------------------------	----	--------	----------------	--

name of optical loading image

## RadMon

<b>withradmon</b>	no	boolean	N	Y/N
-------------------	----	---------	---	-----

access EPIC Radiation Monitor files (ECX, ESX) ?

<b>runradmonfix</b>	no	boolean	Y	Y/N
---------------------	----	---------	---	-----

**radmonfix**: run radmonfix (if withradmon=Y)?

## atthkgen

<b>runatthkgen</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

**atthkgen**: run atthkgen?

<b>timestep</b>	no	real	1.0	>0.0s
-----------------	----	------	-----	-------

**atthkgen**: Duration (in sec) of 'step' through attitude quality information

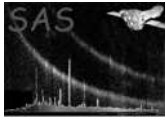
## epframes

<b>runepframes</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

**epframes**: run epframes?

<b>withsrccoords</b>	no	boolean	N	Y/N
----------------------	----	---------	---	-----

whether to use user-supplied values for RA and DEC for TIMING and BURST modes, default is N. See the **warning** in **epframes** (Sect. ??)



<b>srcra</b>	no	angle	0.0	0 .. 360
--------------	----	-------	-----	----------

source right ascension (J2000)

<b>srcdec</b>	no	angle	0.0	-90 .. +90
---------------	----	-------	-----	------------

source right ascension (J2000)

<b>withsrcrawy</b>	no	boolean	N	Y/N
--------------------	----	---------	---	-----

whether to use user-supplied value for source RAWY position for TIMING and BURST modes, default is N, if both `withsrccoords=N` and `withsrcrawy=N` then the RA\_OBJ and DEC\_OBJ from the ProposalInfo are used. See the **warning** at `withsrccoords`!

<b>srcposition</b>	no	integer	190	1-200
--------------------	----	---------	-----	-------

**epframes**: source position for TIMING and BURST mode (in RAWY pixel coordinates)

<b>withfinetime</b>	no	boolean	Y	Y/N
---------------------	----	---------	---	-----

whether to apply fine time correction for TIMING and BURST modes by using source RAWY position, default is Y

<b>lowerthreshold</b>	no	integer	20	0-4095
-----------------------	----	---------	----	--------

disregard low-energy events (with amplitudes < lowerthreshold [adu]) already at this stage, default lowerthreshold=0 preserves recommended (old) behavior. This may be useful when comparing early mission data with recent observations as the setup was different (lowerthreshold=23 instead of 20 now)

<b>wrongpixlimit</b>	no	integer	10	0-100
----------------------	----	---------	----	-------

**epframes**: allowed percentage of 'wrong' events before sending a warning

<b>mipmethod</b>	no	string	onboard	none,sas,com,onboard
------------------	----	--------	---------	----------------------

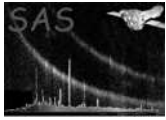
**epframes**: method to handle MIPs in raw events data (mainly for commissioning phase)

<b>qualmax</b>	no	integer	0	
----------------	----	---------	---	--

**epframes**: maximum allowed value of the quality flag to keep event in list (mainly for commissioning phase)

<b>ecntempqb1</b>	no	real	-9999.9	
-------------------	----	------	---------	--

**epframes**: quadrant box temperature [deg C] E.Cn\_TEMPQB1 [F1576 F1676 F1776 F1876], (temporary parameter mainly for calibration, overwritten by existing PAH file values)



<b>f1294</b>	no	integer	0	0-31
--------------	----	---------	---	------

**epframes:** quadrant wait states [F1294], defines the length of the Extended Full Frame Mode frame time, typical values are 0, 3, or 5 (temporary parameter, overwritten by existing summary file values)

<b>f1118</b>	no	string	Unknown	UNKNOWN Open Closed Thin1 Thin2 Medium Thick CalOpen CalClosed CalThin1 CalThin2 CalMedium CalThick
--------------	----	--------	---------	--

**epframes:** filter name (temporary parameter, overwritten by existing summary file values)

<b>anchop</b>	no	integer	0	0-255
---------------	----	---------	---	-------

**epframes:** An\_CHOP [F1534 F1634 F1734 F1834] (temporary parameter, overwritten by existing summary file values)

<b>automode</b>	no	boolean	N	Y/N
-----------------	----	---------	---	-----

try to determine mode from the data itself [not implemented yet]

<b>autofilter</b>	no	boolean	N	Y/N
-------------------	----	---------	---	-----

try to determine filter from the data itself [not implemented yet]

<b>odfok</b>	no	boolean	Y	Y/N
--------------	----	---------	---	-----

**epframes:** assume a correct/complete ODF

<b>hkok</b>	no	boolean	Y	Y/N
-------------	----	---------	---	-----

**epframes:** assume correct/complete HK files

<b>guessdeltap</b>	no	boolean	N	Y,N
--------------------	----	---------	---	-----

whether to estimate the shift of the PN oscillator frequency due to temperature and ageing effects from HK data, could be used to estimate SAS\_JUMP\_TOLERANCE (divide by 6).

<b>showaux</b>	no	boolean	N	Y/N
----------------	----	---------	---	-----

**epframes:** increase internally verbosity for AUX-related output

<b>showccx</b>	no	boolean	N	Y/N
----------------	----	---------	---	-----



**epframes:** increase internally verbosity for CCX-related output

<b>showpmh</b>	no	boolean	N	Y/N
----------------	----	---------	---	-----

**epframes:** increase internally verbosity for PMH-related output

<b>showpah</b>	no	boolean	N	Y/N
----------------	----	---------	---	-----

**epframes:** increase internally verbosity for PAH-related output

<b>ancmcorr</b>	no	integer	512	0-4095
-----------------	----	---------	-----	--------

**epframes:** An.CMCORR F1525 F1625 F1725 F1825

<b>aneamipssel</b>	no	integer	1	0-63
--------------------	----	---------	---	------

**epframes:** An.EAMIPSEL F1536 F1636 F1736 F1836

<b>anmaxmip</b>	no	integer	63	0-4095
-----------------	----	---------	----	--------

**epframes:** An.MAXMIP F1527 F1627 F1727 F1827

<b>anmip</b>	no	integer	3512	0-4095
--------------	----	---------	------	--------

**epframes:** An.MIP F1526 F1626 F1726 F1826

<b>ccfok</b>	no	boolean	Y	Y/N
--------------	----	---------	---	-----

**epframes:** Is this correct/nominal ODF/SDF ?

<b>witheventmap</b>	no	boolean	N	Y/N
---------------------	----	---------	---	-----

**epframes:** Create several event CCD maps ?

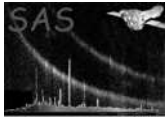
<b>eventmapset</b>	no	file	./eventmap###.dat	
--------------------	----	------	-------------------	--

**epframes:** Name of output event CCD map file

<b>f1052</b>	no	integer	32400	0-32767
--------------	----	---------	-------	---------

**epframes:** DTIMAUTRSTPREVAL coarse time reset [s]

<b>mipdist</b>	no	boolean	N	Y/N
----------------	----	---------	---	-----



**epframes:** Create MIPDIST columns and MIPHIST extension in output ?

<b>mipthreshold</b>	no	integer	3000	0-4095
---------------------	----	---------	------	--------

**epframes:** maximum PHA for non-MIP events [adu]

<b>photonmapset</b>	no	file	./photonmap##.dat	
---------------------	----	------	-------------------	--

**epframes:** Name of output photon CCD map file

<b>setupbpx</b>	no	string	nom6	cal4/nom0/nom1/ nom2/nom3/nom4/ nom5/nom6/none
-----------------	----	--------	------	--

setup for badpix/offset correction vector (used only if ccok=N)

<b>showve</b>	no	boolean	N	Y/N
---------------	----	---------	---	-----

**epframes:** Display valid event intervals ? (not in use yet)

### badpixfind

<b>runbadpixfind</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**badpixfind:** run badpixfind?

<b>badpixset</b>	no	string	./bpxf_##.fits	
------------------	----	--------	----------------	--

**badpixfind, badpix:** path of bad pixel set, the substring ## will internally be replaced with the CCD number currently in use (two digits)

### badpix

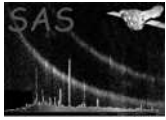
<b>runbadpix</b>	no	boolean	Y	Y/N
------------------	----	---------	---	-----

**badpix:** run badpix?

<b>getuplnkbadpix</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----

**badpix:** get uplinked bad pixels (from CCF)?

<b>getotherbadpix</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----



**badpix:** get non-uplinked bad pixels (from CCF)?

<b>getnewbadpix</b>	no	boolean	Y	Y/N
---------------------	----	---------	---	-----

**badpix:** get new pixels from task **badpixfind**? Not applicable (active) for TIMING and BURST mode

<b>emptyextension</b>	no	boolean	N	Y/N
-----------------------	----	---------	---	-----

**badpix:** create an empty BADPIX extension?

<b>windowfilter</b>	no	boolean	N	Y/N
---------------------	----	---------	---	-----

**badpix:** just get pixels within input file X/Y window?

## eproject

<b>runeproject</b>	no	boolean	N	Y/N
--------------------	----	---------	---	-----

**eproject:** run eproject?

<b>badcolumnset</b>	no	string	badcolumn.tab	
---------------------	----	--------	---------------	--

optional bad column list (ascii)

<b>sigma</b>	no	real	4.0	
--------------	----	------	-----	--

sigma threshold for offset correction

<b>noiseparameters</b>	no	13 × real	0.98 12 × 1.0	
------------------------	----	-----------	---------------	--

noise fraction parameters (cutoff parameter and 12 chip specific correction factors; only for expert use)

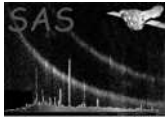
<b>withoffsetlist</b>	no	boolean	no	
-----------------------	----	---------	----	--

enables use of list of offset maps to calculate energy shifts

<b>odilist</b>	no	dataset	odilist.asc	
----------------	----	---------	-------------	--

Name of optional ASCII file containing pairs of <ccd nr.> <offset map file> (one per line). See Sect.3.6.

<b>withxrlcorrection</b>	no	boolean	N	Y/N
--------------------------	----	---------	---	-----



execute X-ray loading correction code (for TI+BU modes) ?

<b>withsoftflarescreening</b>	no	boolean	N	Y/N
-------------------------------	----	---------	---	-----

execute soft flare screening code (for TI mode) ?

<b>softflarethreshold1</b>	no	real	10.0	
----------------------------	----	------	------	--

threshold 1 for flare screening (unit: counts/0.1 s)

<b>softflarethreshold2</b>	no	real	1.0	
----------------------------	----	------	-----	--

threshold 2 for flare screening

<b>softflaresmooth</b>	no	string	BOX	BOX GAUSS FLARE
------------------------	----	--------	-----	-----------------

smoothing method for flare screening

<b>softflareenergyrange</b>	no	2 × integer	40 50	
-----------------------------	----	-------------	-------	--

energy range for flare screening (ADU)

<b>softflaresmoothparams</b>	no	real	2.0 1.0 1.0	
------------------------------	----	------	-------------	--

smoothing parameters

## epnoise

<b>runepnoise</b>	no	boolean	N	Y/N
-------------------	----	---------	---	-----

epnoise: run epnoise?

<b>identifynoisyframes</b>	no	boolean	Y	Y/N
----------------------------	----	---------	---	-----

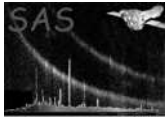
Identify Noisy Frames

<b>sigmacut</b>	no	real	3.0	
-----------------	----	------	-----	--

sigma cut for bright sources

<b>applyfilter</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

Keep output of filtering process?



<b>savemasks</b>	no	boolean	N	Y/N
------------------	----	---------	---	-----

Save CCDs mask to a file

## epxrlcorr

<b>runepxrlcorr</b>	no	boolean	N	Y/N
---------------------	----	---------	---	-----

**epxrlcorr**: run epxrlcorr?

This task (for IMAGING modes) requires an offsetmap to be present in the ODF, which was generally not the case in the first years (it would stop then).

<b>withxrlimage</b>	no	boolean	N	Y/N
---------------------	----	---------	---	-----

**epxrlcorr**: create a diagnostic X-ray loading image?

## epevents

<b>runepevents</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

**epevents**: run epevents?

<b>withphotonmap</b>	no	boolean	N	Y/N
----------------------	----	---------	---	-----

**epevents**: whether to create diagnostic photon map file with 4 image extensions

<b>reemissionthresh</b>	no	integer	0	
-------------------------	----	---------	---	--

**epevents**: selection parameter: trigger threshold (in adu) for preceding events

<b>randomizeposition</b>	no	boolean	Y	Y/N
--------------------------	----	---------	---	-----

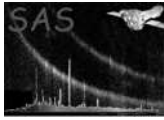
**epevents**: yes, if the computation of physical camera detector coordinates is done with randomization

<b>randomizeenergy</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

**epevents**: yes, if the raw amplitudes should be randomized within a pulseheight bin

<b>gainctiaccuracy</b>	no	integer	2	0-2
------------------------	----	---------	---	-----

**epevents**: Accuracy of gain/cti correction



<b>patternanalysis</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

**epevents:** no, if pattern recognition has been done already (future development)

<b>withoutoftime</b>	no	boolean	N	Y/N
----------------------	----	---------	---	-----

**epevents:** yes, if “out-of-time events” file should be created instead of “normal events” file (only meaningful for IMAGING modes) (considered as *experimental*)

<b>withctisrcpos</b>	no	boolean	N	Y/N
----------------------	----	---------	---	-----

**epevents:** yes, if not the RAWY coordinates but the source position SRCPOS should be used in the energy correction routines (only meaningful for IMAGING modes) (considered as *experimental*). See Sect.3.2.

<b>withbackgroundgain</b>	no	boolean	Y	Y/N
---------------------------	----	---------	---	-----

**epevents:** yes, if background gain corrections should be applied

<b>backgroundtres</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----

yes, if time resolved (rather than averaged discarded line value) shall be used for background gain corrections

<b>backgroundtbin</b>	no	real	100.0	0-150000
-----------------------	----	------	-------	----------

time bin for averaging time-resolved background level data

<b>withpatternoffset</b>	no	boolean	Y	Y/N
--------------------------	----	---------	---	-----

**epevents:** yes, if pattern energy offset corrections should be applied

<b>withctilongterm</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

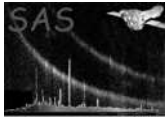
**epevents:** yes, if long-term CTI increase corrections should be applied

<b>ctilongtermsoft</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

**epevents:** yes, if special soft energy function should be included in the long-term CTI increase corrections (considered as *experimental*)

<b>ctilongtermy</b>	no	boolean	Y	Y/N
---------------------	----	---------	---	-----

**epevents:** yes, if special Y dependence should be included in the long-term CTI increase corrections



<b>withccdoffsets</b>	no	boolean	N	Y/N
-----------------------	----	---------	---	-----

**epevents:** yes, if CCD offset corrections should be applied (considered as *experimental*)

<b>withtempcorrection</b>	no	boolean	Y	Y/N
---------------------------	----	---------	---	-----

**epevents:** yes, if temperature-gain corrections should be applied

<b>withgainburst</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**epevents:** apply special gain if BURST mode ?

<b>withgaintiming</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----

**epevents:** apply special gain if TIMING mode ?

<b>withgainff</b>	no	boolean	N	Y/N
-------------------	----	---------	---	-----

**epevents:** apply special gain if FULL FRAME mode ?

<b>withgaineff</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

**epevents:** apply special gain if EXTENDED FULL FRAME mode ?

<b>withphagaincolumn</b>	no	boolean	N	Y/N
--------------------------	----	---------	---	-----

**epevents:** Whether to create intermediate column PHA.GAIN. If set to “Y” then `propagatecolumns` is reset to “all”.

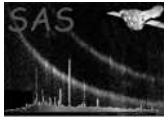
<b>lowgainenergyscale</b>	no	boolean	N	Y/N
---------------------------	----	---------	---	-----

**epevents:** When switching on the energy correction for low-gain mode data then most of the events will fall outside the 2-byte-limit for the PI column (i.e.  $> 32757$  eV) as the energy range is then about 2 – 280 keV; if one is interested in this full range the setting “N” should be used and energy values be multiplied later with 18.4 to obtain “real” event energies. Only effective for the few low-gain mode exposures, of course.

<b>checksasmip</b>	no	boolean	N	Y/N
--------------------	----	---------	---	-----

yes, if the MIP rejection information obtained by task `epframes` shall be printed (only meaningful if on-board rejection is switched off, i.e. for SW, TI, BU modes).

<b>withrdpha</b>	no	boolean	Y	Y/N
------------------	----	---------	---	-----



yes, if a correction for rate-dependent PHA effects for TI and BU modes should be applied. The logical keyword PHA\_RDCO indicates whether this correction has been applied or not. If applied, then the keyword PHA\_RDCB gives the scaling factor B used in the correction, derived from block RDPHA\_DERIV in the CTI.CCF.

<b>rdphatimebinsize</b>	no	boolean	100	
-------------------------	----	---------	-----	--

time-bin size for rate-dependent PHA correction for TI and BU modes [s]

<b>withframecti</b>	no	boolean	N	Y/N
---------------------	----	---------	---	-----

yes, if TIME-derived frame numbers should be used in CTI correction for non-imaging modes (TI, BU) instead of the ODF frame numbers. For FF, eFF, LW, SW modes internally always the TIME-derived frame numbers instead of the dummy ODF numbers are used (should not be changed).

<b>testenergywidth</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

yes, if use non-standard energy bin width (i.e., 1 eV instead of previously used 5 eV binning in output PI column

### attcalc

<b>runattcalc</b>	no	boolean	Y	Y/N
-------------------	----	---------	---	-----

**attcalc:** run attcalc?

<b>attitudelabel</b>	no	string	ahf	ahf/fixeD/om
----------------------	----	--------	-----	--------------

**attcalc:** source of attitude data

<b>refpointlabel</b>	no	string	nom	nom/obj/pnt/user
----------------------	----	--------	-----	------------------

**attcalc:** source of celestial coordinates of central reference point

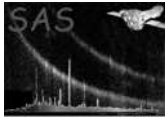
<b>fixedra</b>	no	real		
----------------	----	------	--	--

**attcalc:** S/C's attitude (RA) in the equatorial, earth-centred reference frame (if **attitudelabel**=fixed)

<b>fixeddec</b>	no	real		
-----------------	----	------	--	--

**attcalc:** S/C's attitude (DEC) in the equatorial, earth-centred reference frame (if **attitudelabel**=fixed)

<b>fixedposangle</b>	no	real		
----------------------	----	------	--	--



**attcalc:** S/C's attitude (PA) in the equatorial, earth-centred reference frame (if **attitudelabel=**fixed)

<b>nominalra</b>	no	real		
------------------	----	------	--	--

**attcalc:** celestial coordinate RA of central reference point (if **refpointlabel=**user)

<b>nominaldec</b>	no	real		
-------------------	----	------	--	--

**attcalc:** celestial coordinate DEC of central reference point (if **refpointlabel=**user)

<b>imagesize</b>	no	real	0.36	
------------------	----	------	------	--

**attcalc:** half-size of final image (in degrees)

### time and exposure

<b>runepexposure</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**epexposure:** Run epexposure ?

<b>screenexposure</b>	no	boolean	Y	Y/N
-----------------------	----	---------	---	-----

**epexposure:** remove all columns in EXPOSUnn extensions that are beyond the ICD

<b>spatialexposure</b>	no	boolean	Y	Y/N
------------------------	----	---------	---	-----

**epexposure:** yes, if spatial exposure inhomogeneities (CCD columns) should be determined and to be taken into account by subsequent SAS tasks (not yet implemented)

<b>randomizetime</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**epexposure:** yes, if the event arrival times should be randomized within a readout frame

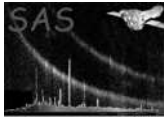
### event screening

<b>screen</b>	no	boolean	Y	Y/N
---------------	----	---------	---	-----

reject all events with rejection flags and below low-energy threshold

<b>runscreen</b>	no	boolean	Y	Y/N
------------------	----	---------	---	-----

perform the screening (otherwise use existing files)



<b>screenlowthresh</b>	no	real	150	
------------------------	----	------	-----	--

reject all events with lower (recombined) energies [eV]

<b>screenrejected</b>	no	boolean	N	Y/N
-----------------------	----	---------	---	-----

reject all events with rejection flags (apply #XMMEA\_EP in screening)

**evlistcomb**

<b>runevlistcomb</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**evlistcomb**: run evlistcomb?

<b>memorymodel</b>	no	string	high	low/highlow/high
--------------------	----	--------	------	------------------

memory model for task **evlistcomb**, all other tasks are controlled via SAS\_MEMORY\_MODEL [current SAS default: high]

<b>withmedianpnt</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

**evlistcomb**: Get median values from **atthkgen** file (otherwise mean)?

<b>othertables</b>	no	list of upper-case strings	BADPIX EXPO-SURE	none
--------------------	----	----------------------------	------------------	------

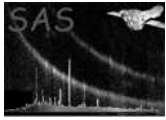
**evlistcomb**: names of secondary tables or arrays to propagate

<b>epningcolnames</b>	no	list of upper-case strings	TIME RAWX RAWY DETX DETY X Y PHA PI FLAG PATTERN PAT.ID PAT_SEQ	none
-----------------------	----	----------------------------	---	------

**evlistcomb**: columns to propagate in main table (IMAGING mode)

<b>epningcoltypes</b>	no	list of strings	double int16 int16 int16 int16 int32 int32 int16 int16 int32 int8 int16 int8	int8/int16/int32/single/ double/boolean/string
-----------------------	----	-----------------	---	---

**evlistcomb**: output type of IMAGING mode columns



<b>epntimcolnames</b>	no	list of upper-case strings	TIME RAWX RAWY DETX DETY X Y PHA PI FLAG PATTERN PAT_ID PAT_SEQ	none
-----------------------	----	----------------------------	---	------

**evlistcomb**: columns to propagate in main table (TIMING mode)

<b>epntimcoltypes</b>	no	list of strings	double int16 int16 int16 int16 int32 int32 int16 int16 int32 int8 int16 int8	int8/int16/int32/single/ double/boolean/string
-----------------------	----	-----------------	---	---

**evlistcomb**: output type of TIMING mode columns

<b>propagatecolumns</b>	no	string	auto	auto/imaging/subset/ icd/all
-------------------------	----	--------	------	---------------------------------

how many columns should be propagated to the final event file for TIMING and BURST modes: “auto” means values via **epntimcoltypes** and **epntimcolnames**, “imaging” uses the values as for **epnimgcolnames** and **epnimgcoltypes**, “subset” similar to “imaging” but without X, Y, and PAT\_SEQ, whereas “icd” uses only the columns that are explicitly mentioned in the Data Products ICD, and “all” uses all output columns of **epevents** for IMAGING and TIMING modes

### final GTI screening

<b>runevselect</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

**evselect**: run evselect?

### RDCTI correction

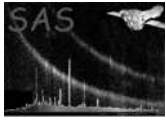
<b>runepfast</b>	no	boolean	N	Y/N
------------------	----	---------	---	-----

run **epfast** for rate-dependent CTI (RDCTI) corrections in TI+BU modes? Ignored if **withrdpha**="Y" or if IMAGING mode.

### pattern distribution

<b>withpatplot</b>	no	boolean	N	Y/N
--------------------	----	---------	---	-----

run **epatplot** to create pattern plot and FLAG=0 masks



<b>modifyinset</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

if N: skip writing of header keywords to increase speed

## background lightcurve

<b>runbackground</b>	no	boolean	Y	Y/N
----------------------	----	---------	---	-----

create background lightcurve?

<b>withmask</b>	no	boolean	Y	Y/N
-----------------	----	---------	---	-----

use masks for lightcurve?

<b>maskset</b>	no	string	./mask_##.fits	
----------------	----	--------	----------------	--

path of point source mask set, the substring ## will internally be replaced with the CCD number currently in use (two digits). Energy range: 0.2-10.0 keV. Not applicable (active) for SMALL\_WINDOW, TIMING, and BURST mode

<b>withhardmask</b>	no	boolean	Y	Y/N
---------------------	----	---------	---	-----

use hard energy band for maskset instead of soft ?

<b>hrdmskset</b>	no	string	./hrdmask_##.fits	
------------------	----	--------	-------------------	--

same as above but for energy range 7.0-15.0 keV

<b>rateset</b>	no	string	rate_bkg.fits	
----------------	----	--------	---------------	--

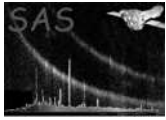
name of background lightcurve file

<b>specset</b>	no	string	spec_bkg.fits	
----------------	----	--------	---------------	--

name of background spectrum file

<b>timebinsize</b>	no	real	0.0	
--------------------	----	------	-----	--

time bin for background lightcurve [s], if set to 0 then the task chooses default values according to the instrument mode and energy range: **withhardmask=Y** 100 [s] for FF eFF modes, 200 for LW, 350 for SW, and 100 for TI BU modes, **withhardmask=N** 10 [s] for FF eFF modes, 20 for LW, 350 for SW, and 10 [s] for TI BU modes



## 5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

**odf** (*error*)

SAS\_ODF environment variable does not exist and no data directory given

**indir** (*error*)

given data directory could not be found

**odfiles** (*error*)

no valid file found for given specifications (data mode, exposure, etc.)

**badpixset** (*error*)

parameter does not contain CCD number place-marker ##

**refpoint** (*error*)

no values given for RA, DEC in the case of `refpointlabel=user`

**attitude** (*error*)

no values given for RA, DEC, PA in the case of `attitudelabel=fixed`

**ccdlist** (*error*)

list of CCDs cannot be expanded, e.g. “2-4-6” or “3,-5”

**ccd** (*error*)

CCD number out of range [1-12]

**exposure** (*error*)

exposure not found in ODF directory

**constituent** (*error*)

one (or more) of the constituent calls ended in error

**SAS\_CCF** (*warning*)

ccf.cif or environment variable SAS\_CCF not found

*corrective action:* continue, check SAS\_CCF environment variable

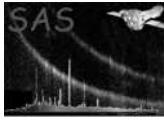
**notInODF** (*warning*)

requested CCD file seems to be non-existent in ODF, skip processing for this CCD

*corrective action:* continue, check ODF or give other CCD list parameter when reprocessing

## 6 Input Files

1. `badpixset` with ## replaced by CCD number: bad pixel files (one per CCD, extension `BADPIX`) (if `getnewbadpix` set to “true” and `runbadpixfind` to “false”)
2. event list files (one per CCD and exposure), straight from the ODF (`PNIME1` or `PNTIE1` or `PNBUE1`, depending on `datamode`)
3. corresponding auxiliary (`PNAUX1`, `PNAUX2`), counting cycle report (`PNCCX1`), housekeeping (`PNPMH1`, `PNPAH1`), and attitude history (`SCATS1`) files, straight from the ODF. Time correlation (`SCTCS1`) file and summary file are accessed by OAL.



## 7 Output Files

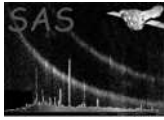
In the case of `usenononicalnames=N` the output files have fixed names or can be given via parameter settings. If `usenononicalnames=Y` then the task creates the file names automatically according to the ODF, where `oooooooooo` denotes the 10-digit observation identifier and `PNxnnn` is the exposure identifier with `x` being the schedule flag (S or U) and `nnn` the 3-digit exposure number.

- merged and calibrated event list file (one per exposure, `EVENTS`), as defined in the Data Products ICD ([1])<sup>1</sup> with CCD specific bad pixel (`BADPIXnn`), exposure (`EXPOSUnn`), GTI extensions (`STDGTInn`), as well as `OFFSETS` with all columns with a non-zero offset and `CALINDEX` with all relevant CCF entries (`EPN`, `XRT3`, `XMM`).  
If `usenononicalnames=Y`: depending on the setting of `withoutoftime` and `withctisrcpos`:  
NN: `PooooooooooPNxnnnPIEVLI0000.FIT` (imaging modes, i.e. FF, eFF, LW, SW) or  
`PooooooooooPNxnnnTIEVLI0000.FIT` (fast modes, i.e. TI, BU)  
YN: `PooooooooooPNxnnnOEVLI0000.FIT`  
NY: `PooooooooooPNxnnnPSEVLI0000.FIT`  
YY: `PooooooooooPNxnnnOSEVLI0000.FIT` (note that this setting is possible but not very meaningful ;-)
- `atthk.dat`: output file from `atthkgen` containing the entire attitude information (`ATTHK`).  
If `usenononicalnames=Y`: `PooooooooooOBX000ATTTSR0000.FIT`
- `gti##.dat` (with `##` replaced by CCD number): CCD specific GTI files (`STDGTInn`)
- `mask_##.fits`: CCD specific (source-free) background masks with `##` replaced by CCD number (created by `badpixfind` if `runbackground` set to “true”).  
If `usenononicalnames=Y`: `PooooooooooPNxnnnBPFMSK00##.FIT`
- `bpxf_##.fits`: CCD specific bad pixel list with `##` replaced by CCD number (created by `badpixfind` if `runbadpixfind` set to “true”).  
If `usenononicalnames=Y`: `PooooooooooPNxnnnBPXFLI00##.FIT`
- `eventmap##.dat` (with `##` replaced by CCD number): CCD event intensity maps (if `witheventmap=Y`), for details see `epframes`
- `photonmap##.dat` (with `##` replaced by CCD number): CCD photon intensity maps (if `withphotonmap=Y`), for details see `epevents`
- `flag0_map_##.dat` (with `##` replaced by CCD number): CCD maps with regions `FLAG=0` set to 1 (if `withpatplot=Y`), for details see `epatplot`.  
If `usenononicalnames=Y`: `PooooooooooPNxnnnFLGMSK00##.FIT`
- value of parameter `rateset`: background lightcurve using source masks with columns `COUNTS` and `RATE` where the latter is scaled by `timebinsize` and the sum of all non-masked pixels, in units of `cts/ks/arcmin2`. There is also a GTI file created with name `bkg.GTI.fits`.  
If `usenononicalnames=Y`: `PooooooooooPNxnnnFBKTSR0000.FIT` for the lightcurve and  
`PooooooooooPNxnnnFBKGTI0000.FIT` for the GTI file.

## 8 Intermediate Files

- `rawevents##.dat`: re-formatted EPIC pn ODF event lists, created by `epframes`, with `##` replaced by CCD number

<sup>1</sup>For TIMING and BURST modes the columns `RAWY` and `PAT.ID` are produced beyond the current version of the ICD.



2. events##.dat.dat: calibrated event lists produced by **epevents**, with ## replaced by CCD number
3. cleanevents##.dat.dat: calibrated event lists produced by **evselect**, with ## replaced by CCD number if **runscreen=Y**

## 9 Algorithm

```
#!/usr/local/bin/perl -w
use strict;

$parameters = default_values
$parameters = command_line_values

atthkgen atthkset=$atthkset timestep=$timestep

for $ccd (@ccdlist) {

    epframes set='$set' eventset=$outfile0 \
        gtiset=$outgti wrongpixlimit=$wrongpixlimit \
        srcposition=$srcposition mipthreshold=$mipthreshold \
        mipmethod=$mipmethod qualmax=$qualmax \
        witheventmap=$witheventmap

    if ($runbadpixfind == "y") {
        badpixfind eventset=$outfile0 badpixset=$bad
    }

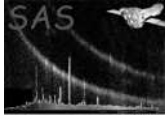
    badpix eventset=$outfile0 badpixset=$bad \
        getuplnkbadpix=$getuplnkbadpix \
        getotherbadpix=$getotherbadpix \
        getnewbadpix=$getnewbadpix \
        emptyextension=$emptyextension \
        windowfilter=$windowfilter

    epevents eventset=$outfile1 outset=$outfile2 \
        reemissionthresh=$reemissionthresh \
        randomizeenergy=$randomizeenergy \
        randomizeposition=$randomizeposition \
        gainctiaccuray=$gainctiaccuracy \
        withphotonmap=$withphotonmap

    attcalc eventset=$outfile2 attitudelabel=$attsou \
        refpointlabel=$refpoint atthkset=$atthkset \
        withmedianpnt=$withmedianpnt imagesize=$imagesize \
        [ fixedra=$attra fixeddec=$attdec fixedposangle=$attPA ]
        [ nominalra=$nominalra nominaldec=$nominaldec ]
}

SAS_MEMORY_MODEL = $memorymodel

evlistcomb eventsets=\"$evlist\" instrument=epn imagingset=$imagingset \
    timingset=$timingset othertables=$othertables
```



```
evselect table=[$imagingset $timingset] expression=\"\${evselexpr}\" \  
          withfilteredset=Y keepfilteroutput=Y filteredset=$outfil \  
          writedss=Y updateexposure=N destruct=Y
```

```
SAS_MEMORY_MODEL = original_value
```

## 10 Comments

- Parameter `odf` takes precedence over `SAS_ODF`.
- `SAS_VERBOSITY` should be set to 4 or 5 for normal processing.
- `SAS_MEMORY_MODEL` should be set to 'high' for normal processing.
- The current implementation is a Perl script. It is not yet fully embedded in SAS and thus does not support all SAS task options.  
`epchain --v` (or `-v`) lists the version number of all chain tasks.  
`epchain --p` (or `-p`) gives a list of all available parameters with its default values.

## 11 Future developments

The chain will adapt to the evolution of its constituents and to the organisation of the pipeline.

## References

- [1] SSC. XMM Survey Science Centre to Science Operations ICD for SSC Products. Technical Report XMM-SOC-ICD-0006-SSC Issue 2.1, SSC, Mar 2000.