epframes

August 15, 2005

Abstract

This task reformats the EPIC ODF PN events list into the file format required by subsequent off-line analysis. **epframes** converts frame numbers to event specific time tags. It determines invalid events and frames and creates corresponding GTI files per CCD. In addition, **epframes** appends the CCD specific exposure extension to the output event list. Optionally, auxiliary information for cosmic ray analysis is collected.

1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING, TIMING, BURST

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

3.1 General

epframes reads a raw event list of a single CCD and exposure from an EPIC PN ODF ([1]) [1]. The input file is specified giving the actual filename or a symbolic form of it (which will be expanded with OAL_expandFileName). The ODF in use is defined via the SAS environment variable `SAS_ODF` prior to execution. **epframes** uses the summary file from the ODF to display info about the selected observation and to fill in standard header keywords. The auxiliary, main periodic housekeeping and spacecraft time correlation files corresponding to the selected EPIC PN raw event list are needed for the computation of event time tags.

[1] Note, that according to **oal** the *exposure number* is the actual running number of the exposure in that particular instrument mode, e.g. OAL(EPN, IMAGING, 14, 11) means the task is looking for CCD 11 of the 14th imaging mode exposure of EPIC-pn in the ODF while ODF(EPN, IMAGING, 2, 11, U) stands for an unscheduled observation with exposure identifier PNU002.

epframes flags events with faulty coordinates or times in a `FLAG` column. The flags are bit-masked following the same code as for the MOS (see [5]). **epframes** stores statistics about these wrong events in FITS keywords (`WRGCOORD`, `WRGORDER` and `WRGTIME`) in the output event list. The statistics of bad frames is stated in the history of the output file.

The raw event list is converted into a table to be processed by other SAS tasks: The existing columns `RAWX`, `RAWY` and `ENERGY` in the ODF files are renamed to `RAWX`, `RAWY` and `PHA`, respectively. We note that the RAWX and RAWY coordinates in the output file have values increased by +1 with respect to the raw coordinates in the ODF/SDF file. For the same reason **epframes** adds 1 to the `WINDOWXO`, `WINDOWYO` attributes giving the bottom left corner of the readout window. The `PHA` values are the event raw amplitudes corrected for column offsets. A `CCDNR` column gives the overall CCD number (1 - 12).

The `FRAME` counter column (in TIMING and BURST mode together with the `RAWY` events coordinate and the `srcposition` parameter or in the case of `withsrccoords`=Y `srcra` and `srcdec` instead) is used to compute the `TIME` for each event. The time tag computation is based on equations given in EPIC-EST-OP-002 Sects.8.3.1 and 8.3.2 (version Sept. 99) [4]. Please, see the **warning** in the description of the parameter `withsrccoords`!

A GTI file is produced listing start and stop times of the observation and taking care of missing or bad frames.

epframes creates an EXPOSURE extension (renamed by **evlistcomb** to EXPOSUxx, xx = 01 - 12) giving details about frame time tags and integration times. These have been corrected for mode dependent readout gaps (livetime factor FC0).

3.2 Diagnostic images and columns

If `witheventmap`=Y then **epframes** creates a a file (`eventmapset`) with 4 image extensions with all events, only MIP events, soft and hard single events after MIP rejection, respectively.

If `mipmethod` is set to "none" then MIP events are only flagged but not the column of this event in this readout frame. if set to "sas" then also the corresponding column is flagged, and "com" means additional flagging of the neighbouring columns with flagging of the whole frame if five or more MIPs in one frame. "onboard" emulates the revised algorithm after the onboard software update which is similar to "com" but without automatic full frame rejection. `mipthreshold` should not be changed at the moment. `qualmax` can be used to suppress garbage data that has escaped onboard rejection from further processing. *This parameter will be removed soon.* `mipdist` controls whether to create columns `MIPDISTR` and `MIPDISTL` (distance of event to next MIP in same frame to the right and to the left, respectively) in `EVENTS` extension and histogram extension `MIPDHIST` (possible distribution of distances to next MIP).

3.3 Discarded lines and columns

If the onboard MIP rejection is switched on (FF, eFF, LW modes), then the number of rejections of a CCD row (if row-wise MIP rejection) or CCD column (if column-wise MIP rejection, default) integrated over an exposure is stored in a DLI file for this CCD. For the faster (SW, TI, BU) modes the onboard MIP rejection is switched off and no DLI files are present in the ODF.

For all pn modes the task **epframes** performs a MIP rejection on the data to identify residual MIPs not rejected onboard. The number of rejected columns/rows found in the ODF and by the SAS are stored in an extension `DLIMAP` with columns `DLIODF` and `DLISAS`, respectively, for subsequent analysis in later tasks in the pn processing chain.

3.4 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 three equivalent calls to epframes:

```
epframes set=0084_0099280101_PNS01804IME.FIT eventset=r04.dat gtiset=g04.dat
epframes set='OAL(EPN,IMAGING,2,4)'          eventset=r04.dat gtiset=g04.dat
epframes set='ODF(EPN,IMAGING,18,4,S)'       eventset=r04.dat gtiset=g04.dat
```

This processes CCD 4 of the second IMAGING mode exposure (i.e. PNS018). Diagnostic output images can be created with the additional setting

```
epframes ... witheventmap=Y eventmapset=m04.dat
```

and you can adjust the energy bands with parameters

```
epframes ... witheventmap=Y eventmapset=m04.dat \
  nomipthresholdhigh=... nomipthresholdlow=...  \
  softthresholdhigh=...  softthresholdlow=...
```

If you suspect that your source is not at the nominal position but offset then you may enter for TIMING and BURST modes to get the event times (and later CTI correction in task **epevents**) correctly:

```
epframes ... withsrccoords=Y srcra=... srcdec=...
```

If you are interested in the setup of the instrument it is suggested to enable the output of additional messages (if `SAS_VERBOSITY` = 4 or higher), see also Sect.7.2:

```
epframes ... showpah=Y showpmh=Y
```

If you have old PMS (odffixed) data you may wish to set

```
epframes ... odfok=N hkok=N
```

which eliminates a couple of warnings related to missing information in the files that are included and expected in correct (ODS) ODF datasets (this setting should *not* be used in the GO phase).

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

General parameters

set	yes	data-set	odfset	

input EPIC PN ODF (actual or symbolic) data-set name

eventset	yes	data-set	rawevents.dat	

output EPIC PN raw events data-set

gtiset	yes	data-set	gti.dat	

output GTI data-set name

withsrccoords	no	boolean	N	Y,N

whether to use user-supplied values for RA and DEC for TIMING and BURST modes, default is N. If `withsrccoords`=Y then `withsrcrawy` and `srcposition` are ignored. If both `withsrccoords`=N and `withsrcrawy`=N then the RA_OBJ and DEC_OBJ from the ProposalInfo are used as input position.
Warning: For an offset pointing (there may be scientific reasons for this) the values RA_OBJ and DEC_OBJ do *not* represent the desired source position but just the pointing position, you *have* to set `withsrccoords`=Y for this case and supply the proper source coordinates via `srcra` and `srcdec` or to provide the pixel position via `withsrcrawy`=Y and `srcposition`, otherwise the absolute timing and the energy corrections will be incorrect. There *is no other way* to handle this offset case, you have been warned. This is irrelevant for all IMAGING modes.

srcra	no	angle	0.0	0 .. 360

source right ascension (J2000) for `withsrccoords`=Y

srcdec	no	angle	0.0	-90 .. +90

source right ascension (J2000) for `withsrccoords`=Y

withsrcrawy	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`!

srcposition	no	integer	190	1-200; 181-200 should be used for BURST mode

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

withfinetime	no	boolean	Y	Y,N

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

lowerthreshold	no	integer	0	0-4095

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

withfifogti	no	boolean	N	Y,N

whether to search for short duration FIFO overflow data gaps (in preparation)

fifogtithresh	no	real	76.0	0-4095

threshold for FIFI GTI search (in preparation)

witheventmap	no	boolean	N	Y,N

Create several event CCD maps ? These may be useful as they visualize the raw data without any corrections.

eventmapset	no	data-set	eventmap.dat	

Name of output event CCD map fileoutput GTI data-set name

nomipthresholdlow	no	integer	40	0-4095

minimum PHA value for non-MIP events to be included in `NOMIPMAP`, cf. `witheventmap`

nomipthresholdhigh	no	integer	1600	0-4095

maximum PHA value for non-MIP events to be included in `NOMIPMAP`, cf. `witheventmap`

softthresholdlow	no	integer	20	0-4095

minimum PHA value for non-MIP events to be included in `SOFTMAP`, cf. `witheventmap`

softthresholdhigh	no	integer	30	0-4095

maximum PHA value for non-MIP events to be included in `SOFTMAP`, cf. `witheventmap`

Diagnostic and debug parameters

wrongpixlimit	no	integer	10	0-100

allowed percentage of wrong events before stopping to send detailed messages

showccx	no	boolean	N	Y,N

whether to internally increase the verbosity for CCX related information

showaux	no	boolean	N	Y,N

whether to internally increase the verbosity for AUX related information

showpmh	no	boolean	N	Y,N

whether to internally increase the verbosity for PMH related information

showpah	no	boolean	N	Y,N

whether to internally increase the verbosity for PAH related information

showve	no	boolean	N	Y,N

whether to internally increase the verbosity for subroutine ValidEvents related information, optimization critical

MIP rejection (commissioning phase)

mipthreshold	no	integer	3000	0-4095

maximum PHA value for non-MIP events (mainly for commissioning phase and calibration)

mipmethod	no	string	onboard	none onboard sas com

method to flag MIPs in raw data: none = flag only events but not discarded column, sas = flag events and discarded column, com = flag events and discarded column and neighbouring column, onboard = similar to 'com' but without full frames rejection (mainly for commissioning phase and calibration)

mipdiscard	no	boolean	Y	Y,N

whether to discard flagged MIP events from the output list (for check of fast instrument modes or in case of no onboard rejection, *not active yet*)

qualmax	no	integer	0	0-2100000000

maximum value of quality flag to include event in output list (mainly for commissioning phase and cali-

bration). *This parameter will be removed, use* `mipdiscard` *then.*

mipdist	no	boolean	N	Y,N

whether to create columns `MIPDISTR` and `MIPDISTL` in `EVENTS` extension and histogram extension `MIPDHIST` (mainly for commissioning phase and calibration)

PMSFITS parameters (not relevant for GO phase)

f1294	no	integer	0	0-15

Quadrant wait states [F1294], defines the length of the Extended Full Frame Mode frame time, typical values are 0 (for all modes except eFF), 3, or 5 (will be removed once complete PAH and PMH files in ODF available)

f1052	no	integer	32400	0-32767

HK value of DTIMAUTRSTPREVAL [F1052], (will be removed once complete PAH and PMH files in ODF available)

f1118	no	string	UNKNOWN	UNKNOWN Closed Thin1 Thin2 Medium Thick Open CalClosed CalThin1 CalThin2 CalMedium CalThick CalOpen

HK value of filter wheel [F1118], (will be removed once complete PAH and PMH files in ODF available)

anchop	no	integer	0	0-255

An_CHOP [F1534 F1634 F1734 F1834], (will be removed once complete PAH and PMH files in ODF available)

ancmloth	no	integer	532	0-4095

An_CMLOTH [F1515 F1516 F1517 F1615 F1616 F1617 F1715 F1716 F1717 F1815 F1816 F1817], (will be removed once complete PAH and PMH files in ODF available)

ancmcorr	no	integer	512	0-4095

An_CMCORR [F1525 F1625 F1725 F1825], (will be removed once complete PAH and PMH files in ODF available)

aneamipsel	no	integer	1	0-63

An_EAMIPSEL [F1536 F1636 F1736 F1836], (will be removed once complete PAH and PMH files in ODF available)

anmaxmip	no	integer	63	0-4095

An_MAXMIP [F1527 F1627 F1727 F1827], (will be removed once complete PAH and PMH files in ODF available)

anmip	no	integer	3512	0-4095

An_MIP [F1526 F1626 F1726 F1826], (will be removed once complete PAH and PMH files in ODF available)

anmaxfrc	no	integer	20	0-63

An_MAXFRC [F1538 F1638 F1738 F1838], (will be removed once complete PAH and PMH files in ODF available)

ansendmode	no	string	MIP CORREC.2	"NO MIP CORR." "MIP CORREC.1" "MIP CORREC.2"

An_SENDMODE [F1535 F1635 F1735 F1835], (will be removed once complete PAH and PMH files in ODF available)

ecntempqb1	no	real	-9999.9	

quadrant box temperature [°C]: E_Cn_TEMPQB1 [F1576 F1676 F1776 F1876], (will be removed once complete PAH and PMH files in ODF available)

hcettmpfpf	no	real	-90	

CCD fine temperature [°C]: H_CE_TTMPFPFE [F1128], only active in absence of real PMH files (will be removed once complete PAH and PMH files in ODF available)

automode	no	boolean	N	Y,N

try to determine mode from the data itself

autofilter	no	boolean	N	Y,N

try to determine filter from the HK files itself

odfok	no	boolean	Y	Y,N

Is ODF/SDF correct/nominal (yes/no) (for PMSFITS data setting "N" prevents some warnings)

hkok	no	boolean	Y	Y,N

Is PMH/PAH correct/nominal (yes/no) (for PMSFITS data setting "N" prevents some warnings)

ccfok	no	boolean	Y	Y,N

Is CAL/CCF correct/nominal (yes/no) (for commissioning phase setting "N" enables hard-coded offset shifts as there are no correct CCFs yet for this period; `setupbpx` *must* (!) then be correctly set)

setupbpx	no	string	nom3	none nom0 nom1 nom2 nom3 nom4 cal4 nom5 nom6

setup for badpix/offset correction vector, activated by `ccfok`=N (will be removed once complete PAH and PMH and CCF files in ODF available)

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.

illegalIPPVvalue *(error)*
Parameter IPPV has illegal value, check ODF summary file

NoGoodObtValues *(error)*
Time computation failed because no good frames could be identified

InconsistentTimeInfo *(error)*
Time computation is inconsistent, there may be time back-jumps in the data. Submit a report with SAS_VERBOSITY=5.

toomany *(error)*
Number of frames larger than allowed limit

wrongMode *(error)*
Illegal or missing type of data mode or not scientific data mode

changedMode *(warning)*
data mode has been changed: eFF → FF or FF → eFF
corrective action: continue, ODF seems to be inconsistent

CCDlowGain *(warning)*
HK values of this CCD indicate that the mode is not HIGH gain
corrective action: continue, energy scale will be incorrect

singleBoresightBurst *(warning)*
Unusual RAWY event range, dataset may be in SingleBoresight BURST format
corrective action: continue, window and associated flags may be wrong

doubleWindowTiming *(warning)*
Unusual RAWX event range, dataset may be in double-window TIMING format
corrective action: continue, window and associated flags may be wrong

BPTcode *(warning)*
BadPixelTable codes in PAH.FITS and ADUCONV.CCF differ
corrective action: continue, probably the EPN_ADUCONV_####.CCF or the whole CIF is incorrect (too old), amplitudes in columns with offset shifts may be wrong

missingIPPV *(warning)*
Parameter IPPV not found, will set it to default value
corrective action: continue with default value, check ODF summary file

misFrame *(warning)*
Event frame not found in AUX file
corrective action: continue, AUX file may have been created incorrectly

eventsNoAUX *(warning)*
events file non-empty but AUX file empty
corrective action: continue, AUX file may have been created incorrectly

noNGAUX *(warning)*
No good AUX1 intervals left for analysis
corrective action: continue

noAUX1forCCD *(warning)*
No AUX1 entries for this CCD
corrective action: continue

noAUX2forCCD *(warning)*
No AUX2 entries for this CCD
corrective action: continue

PHAoffset *(warning)*
Unusual amplitude range of ODF event list
corrective action: continue, contact task developer

corruptedAUX1 *(warning)*
PNAUX1 extension contains impossible values, not nominal ODF
corrective action: continue, may fail for this CCD or other CCDs later

noPNAUX1 *(warning)*
PNAUX1 extension missing in AUX file, not nominal ODF
corrective action: continue, skip PNAUX1 related code, time and exposure entries incorrect

noPNAUX2 *(warning)*
PNAUX2 extension missing in AUX file, not nominal ODF
corrective action: continue, skip PNAUX2 related code, exposure entries incorrect

noPNPMH1 *(warning)*
PNPMH1 extension missing in housekeeping file, not nominal ODF
corrective action: continue, skip PNPMH1 related code

noPNPAH1 *(warning)*
PNPAH1 extension missing in housekeeping file, not nominal ODF
corrective action: continue, skip PNPAH1 related code

misHKcolumn *(warning)*
column expected in HK file but is absent, not nominal ODF
corrective action: continue, skip analysis for this column

emptyHKrange *(warning)*
no HK values within limits, use all entries in exposure for analysis
corrective action: continue

notHKconstant *(warning)*
HK value is expected to be constant (or limited to small interval) within the exposure but is not
corrective action: continue

HKoffsetmap *(warning)*
No offset map calculation interval could be determined for this observation
corrective action: continue

OffsetMap *(warning)*
HK values indicate that the filter was not the same during intervals of offset map calculation and science exposure or that the offset map calculation interval was not immediately before this exposure
corrective action: continue

illegalODF *(warning)*
The ODF seems to be inconsistent. See subsection **??**
corrective action: continue, some underlying SAS packages may fail as a consequence

wrongFC0 *(warning)*
No mode dependent livetime factor
corrective action: set mode dependent livetime factor FC0 = 1

wrongFTCoarse *(warning)*
Illegal FTCOARSE time value in AUX file
corrective action: give a warning, continue execution

srcpos *(warning)*
BURST mode expects source position in the range 181 to 200
corrective action: give a warning, continue execution

wrongCoord *(warning)*
invalid event coordinates
corrective action: write an `OUT_OF_CCD_WINDOW` flag and continue execution

wrongReadoutOrder *(warning)*
events are in wrong readout order
corrective action: write an `IN_BAD_FRAME` and `IN_SPOILED_FRAME` flag, continue execution

wrongTimeOrder *(warning)*
ODF file not sorted in time
corrective action: write an `IN_BAD_FRAME` and `IN_SPOILED_FRAME` flag, continue execution

wrongMIPcoord *(warning)*
MIP detected with illegal coordinates
corrective action: continue

MIPrejection *(warning)*
MIP CORREC.1 was commanded (row-wise) this mode is not fully calibrated
corrective action: continue, background rejection and CTI correction may be incomplete

tooLongExposure *(warning)*
`TELAPSE` = `TSTOP-TSTART` of this exposure is longer than the whole scheduled observation as specified by `DURATION` = `DATE_END-DATE_OBS`
corrective action: expoure may be longer than scheduled or event times wrong

SubscriptOutOfRange *(warning)*
Size is estimated from TSTOP-TSTART divided by the frame time plus margin; in this exposure this appears to be insufficient and therefore time inconsistencies must have occured.
corrective action: EXPOSURE extension becomes larger than allocated arrays

date2000 *(warning)*
unusual values of DATE-OBS and/or DATE-END
corrective action: give a warning, continue execution

6 Input Files

1. EPIC PN raw event list, depending on data mode: PNIME1 or PNTIE1 or PNBUE1, respectively (ODF file per CCD with FRAME, RAWX, RAWY, ENERGY columns)
2. corresponding EPIC PN Auxiliary file (PNAUX1, PNAUX2)
3. corresponding EPIC PN periodic housekeeping files (PNPMH1, PNPAH1)
4. corresponding EPIC PN counting cycle report auxiliary file (PNCCX1)
5. Time correlation (SCTCS1), attitude (SCATS1, if withsrccoords=Y) and summary files are accessed by the OAL directly

7 Output Files

1. Raw event list with extensions (names may be renamed by subsequent tasks):
   - EVENTS extension with columns TIME, FRAME, RAWX, RAWY, PHA, CCDNR, FLAG
   - EXPOSURE extension with columns TIME, FRACEXP, NDSCLIN1, NDSCLIN2, NDSCLIN3, NDSCLIN4, FRAMELIM
   - OFFSETS extension with columns RAWX, OFFSETX, CCDNR
   - DLIMAP extension with columns DLIODF, DLISAS

   (a) optional: diagnostic columns MIPDISTR, MIPDISTN in EVENTS extension
   (b) optional: diagnostic extension MIPDHIST
2. GTI file with extension STDGTInn and columns START and STOP
3. optional: diagnostic image file eventmapset with extensions MIPMAP, NOMIPMAP, TOTALMAP, SOFTMAP

7.1 Event attributes

epframes creates event quality flags in column FLAG (Int32). It uses the common MOS/pn event related flag code (see [5]) and sets the following event attributes (other flags are set by the task **epevents**):

FLAG

Attribute	Meaning (information)
IN_SPOILED_FRAME	wrong order of events or events not sorted in time
ON_OFFSET_COLUMN	event on column with higher offset
NEXT_TO_OFFSET_COLUMN	event on column next to one with higher offset

Attribute	Meaning (rejection)
IN_BAD_FRAME	event belongs to bad frame
OUT_OF_CCD_WINDOW	event outside the CCD readout window
MIP_ASSOCIATED	events associated with MIP events

7.2 Housekeeping related keywords

The task also issues a number of housekeeping related messages and values and also writes several keywords to the EXPOSURE extension (the information is usually CCD-dependent and it is thus important that it is kept and propagated through the event file processing chain). The messages appear for verbosities of 4 (recommended value for normal processing) and higher, either specified via e.g. environment variable SAS_VERBOSITY=4 or task parameter "-V 4". The keywords are written to the event file regardlessly the verbosity, of course.

- Filter: the filter wheel position is derived from the position sensor potentiometer (housekeeping column F1122 contained in the main periodic housekeeping file PNPMH1, HK mnemonic H_CE_FWSPOT). From the fixed positions (CLOSED, THIN1, THIN2, MEDIUM, THICK, OPEN, approximately separated by 60 degrees) the corresponding calibration positions (e.g. CALCLOSED) are obtained by turning the filter wheel back by about 5 degrees. Verbosity of 4 (or higher) gives the minimum and maximum of this housekeeping parameter during the exposure time and the avarage value together with the translated filter name. These are written to the keywords F1122MIN, F1122MAX, F1122AVE, respectively.
- High Voltage: V_VC_UHVx, x = 0..3 (quadrant number), columns F1198, F1199, F1200, F1201, typical values are -162.4, -163.1, -151.5, -161.5 [V], respectively. Keywords written: UHV_MIN, UHV_MAX, UHV_AVE.
- CCD temperature: H_CE_TTMPFPF, column F1128, typical value -89.97 [°C], keywords F1128MIN, F1128MAX, F1128AVE
- Quadrant box temperature: E_Cx_TEMPQB1, x = 0..3 (quadrant number), columns F1576, F1676, F1776, F1876 in the auxiliary HK file PNPAH1, respectively. The typical values depend on the instrument mode. Deviations by several degrees from the average value (e.g. if one or two RGS instruments are switched off) indicate that the energy correction may be incomplete later because these gain-temperature-effects need recalibration at the moment. For CCD 4 (whole quadrant 1) typical values are 20.8, 20.8, 20.9, 21.0, 22.7, 21.3 [°C] for FF, eFF, LW, SW, TI, BU modes, respectively. Keywords are TQB1_MIN, TQB1_MAX, TQB1_AVE
- Guard ring current: E_Cx_IGRAy, x = 0..3 (quadrant number), y = 0..2 (CCD ID), PNPAH1 column numbers 1571+CCDID+100*Quadrant, keywords IGRA_MIN, IGRA_MAX,IGRA_AVE, there is considerable scatter due to rough digitization ($2 - 4\mu$A), values slightly temperature dependent
- BadPixelTable code: starting in revolution 293 a unique code identifying the uploaded bad pixel table (column offsets and bad pixels) was put into the HK telemetry, keyword BPT_CODE. This code will also be in the header of the corresponding CCF files ADUCONV and BADPIX (none yet as there was no change of these files since then at the time of writing). It provides an independent check whether the correct ADUCONV and BADPIX calibration files are used.
- Offset map calculation: before each exposure an offset map is calculated onboard using the filter and mode of the subsequent science exposure. The calculation is performed for all quadrants simultaneously, first all central CCDs, then the middle CCDs, and then the outer CCDs of each quadrant. The involved parameters are written to keywords of the form OT_*. As only a small fraction of the calculated offset maps is transmitted to ground these quantities may serve as a tool if problems with the offset map calculation are suspected.

- An almost complete list of available HK columns can be obtained from the XMM technical note "XMM-SOC-TN-0040-SSD" ([2])

8 Algorithm

```
subroutine epframes

* Read input parameters:

  *** Open the ODF in directory SAS_ODF
  *** (can also be defined via --odfdir or -o; see taskmain)
  call OAL_odfInfo

  *** Print infos about the proposal
  call OAL_proposalInfo

  *** Specify which exposure and CCD one is interested in
  call OAL_expandFileName(shortfileName,fileName)
  OAL_selectScope(EPN, IMAGING) or TIMING or BURST
  call OAL_setState(fileName)

  *** Print Infos about the exposure
  call OAL_exposureInfo

  *** Select Data from single CCD

  call OAL_selectFile(ccdNr,fileName)

* Declarations for the DAL:

  *** Create empty output set and copy attributes
  outfile = dataSet(stringParameter("outfile"), CREATE)
  call setAttributes(attributable(outfile),attributable(infile))
  *** Write standard primary header keywords
  call setAttribute(outfile,"TELESCOP","XMM",...)
  call setAttribute(outfile,"INSTRUME","EPN",...)
  call setAttribute(outfile,"DETNAM"," ",...)
  call setAttribute(outfile,"FILTER",filterId(OAL_activeFilter()),...)
  if (odfInfos%isSlew) then
    call setAttribute(outfile,"OBS_MODE","SLEW",...)
  else
    call setAttribute(outfile,"OBS_MODE","POINTING",...)
  endif
  call setAttribute(outfile,"OBS_ID", &
       odfinfos%proposalId+odfinfos%observationId,...)
  call setAttribute(outfile,"EXP_ID",expinfos%number,...)
  select case (readoutMode)
  case(0)
   call setAttribute(outfile,"CONTENT","EPIC IMAGING MODE EVENT LIST",...)
   call setAttribute(outfile,"DATAMODE","IMAGING",...)
  case(1)
   call setAttribute(outfile,"CONTENT","EPIC TIMING MODE EVENT LIST",...)
   call setAttribute(outfile,"DATAMODE","TIMING",...)
```

```
  case(8)
   call setAttribute(outfile,"CONTENT","EPIC BURST MODE EVENT LIST",...)
   call setAttribute(outfile,"DATAMODE","BURST",...)
  end select
  call setAttribute(outfile,"ORIGIN","Leicester/SSC",...)
  call setAttribute(outfile,"OBJECT",proposalinfo%targetName,...)
  call setAttribute(outfile,"DATE-OBS",odfinfos%observationStartTime,...)
  call setAttribute(outfile,"DATE-END",odfinfos%observationEndTime,...)
  call setAttribute(outfile,"OBSERVER",proposalinfo%GoName,...)
  *** Write some additional keywords (to be used by attcalc)
  call setAttribute(outfile,"RA_OBJ",proposalinfo%targetRa,...)
  call setAttribute(outfile,"DEC_OBJ",proposalinfo%targetDec,...)
  call setAttribute(outfile,"RA_NOM",proposalinfo%boresightRa,...)
  call setAttribute(outfile,"DEC_NOM",proposalinfo%boresightDec,...)

  *** Create events table extension (one per CCD)
  outccd = addTable(outfile,"EVENTS",ntrow)
  call setAttributes(attributable(outccd),attributable(inccd))
  call setAttribute(outccd,"TSTART" ,...) ! start time of first frame
  call setAttribute(outccd,"TSTOP"  ,...) ! end time of last frame
  call setAttribute(outccd,"TELAPSE",...) ! TSTOP - TSTART in [s]

  call setEventAttribute(outccd,EVATT_IN_SPOILED_FRAME)
  call setEventAttribute(outccd,EVATT_IN_BAD_FRAME)
  call setEventAttribute(outccd,EVATT_OUT_OF_CCD_WINDOW)

  *** Read table columns FRAME, RAWX, RAWY, ENERGY

* Convert frame counter to event time tags:

  col = column(inccd, "FRAME", READ)
  frames => int32Data(col)

  call OAL_frameCounterToObt
  call OAL_obtToTimeTag

  * fine time correction:

  if IMAGING mode then
   call CAL_correctTime(timeTags)
  else  ! TIMING or BURST with source position
   call CAL_correctTime(timeTags, rawY, srcposition)
  endif

* Compute mode dependent livetime factor FC0:

* Create exposure extension

      TIME     = real64Data(column(inccd,"TIME",READ))
      TIMEDEL  = real64Attribute(inccd,"FRMTIME")/1000 * FC0
      FRACEXP  = 1 (real32Data, other livetime factors to be handled
            by higher SAS task)


* Correct for offset shifts
```

```
  call CAL_offsetCorrect

*** determine frame boundaries and prepare the loop
*** read table column RAWX, RAWY and FLAG frame by frame

do frame = firstFrame, lastFrame

   ixx   = RAWX(frame)
   iyy   = RAWY(frame)
   iqual = FLAG(frame)
   itime = TIME(frame)

   * Search for valid events

     *** Check for wrong coordinates
     read window size and position
     if (ixx,iyy) outside this window, than
       call warning "Error in coordinates"
       qual(i) = ior(qual(i), EVATT_OUT_OF_CCD_WINDOW)
                                    ! SINGLE EVENT FLAGGING:
                                    ! OUTSIDE THE READOUT WINDOW

     *** Check for correct order of events
     pix_code_new = iyy * 100 + ixx
     if (pix_code_new <= pix_code_old) then
       call warning "Events are in wrong order"
       qual(i) = ior(qual(i), EVATT_IN_BAD_FRAME)
                             ! WHOLE FRAME FLAGGING
       qual(i) = ior(qual(i), EVATT_IN_SPOILED_FRAME)
                                    ! WRONG ORDER OF EVENTS
     endif
     pix_code_old = pix_code_new

     *** Check for time order
     if frame not sorted in time
       call warning "Frame not sorted in time"
       qual(i) = ior(qual(i), EVATT_IN_BAD_FRAME)
                             ! WHOLE FRAME FLAGGING
       qual(i) = ior(qual(i), EVATT_IN_SPOILED_FRAME)
                                    ! NOT SORTED IN TIME

enddo    ! loop over frames

*** Display statistics about wrong events

    wronglimit = intParameter("wrongpixlimit")
    if (number_of_wrong_events > real(wronglimit)) &
      call warning "More wrong events than wrongpixlimit allows"

*** Add statistics to the history and keywords

*** Convert EPIC PN ODF file into format defined in ICD
  write outfile with columns
      TIME, FRAME, RAWX, RAWY, PHA, CCDNR, FLAG
```

```
    NOTE: RAWX/Y and attributes WINDOWX0/Y0 are increased by 1

  * Write GTIs depending on start/stop time of observation
    and wrong/missing frames

    *** Create empty output set and copy primary standard attributes
    outgti = dataSet(stringParameter("outgti"), CREATE)
    call setAttributes(attributable(outgti),attributable(outfile))

    *** Create GTI extension
    outgticcd = addTable(outgti,"STDGTIxx",ngti) ! where xx is the overall
                                                 ! CCD number (01-12)
    call setAttribute(outgticcd,"HDUCLASS","OGIP",...)
    call setAttribute(outgticcd,"HDUCLAS1","GTI",...)
    call addAttributes(attributable(outgticcd),attributable(outccd))

    for each GTI:
    start(i) = time(firstEvent) - (FRMTIME/1000)/2
    stop(i)  = time(lastEvent)  + (FRMTIME/1000)/2

  end subroutine epframes
```

9 Comments

- Time related attributes:
  - observation related: DATE_OBS and DURATION required by XMM-SOC-ICD-0006-SSC ([3]), write also DATE_END and OBSSTART, OBSSTOP
  - exposure related: DATE-OBS and DATE-END required by XMM-SOC-ICD-0004-SSD ([1]), write also EXPSTART, EXPSTOP
- SAS_VERBOSITY should be set to 'Sparse' (4) for normal processing and 'Verbose' (5) for more detailed processing. Values of 6 and higher may yield quite a number of messages related to rejected events (MIPs) depending on CCD number and instrument mode.
- SAS_MEMORY_MODEL should be set to 'high' for normal processing.
- Information about possible errors in the ODF is needed to decide about additional criteria to flag wrong frames and events: can the ODF still be corrupted by transfer jumps, bit flips, etc. or are these checks done elsewhere?

10 Future developments

The memory usage of the task is not optimized. This could be improved by using temporary datasets.

References

[1] ESA. XMM Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM). Technical Report XMM-SOC-ICD-0004-SSD Issue 2.5, ESA/SSD, June 2000. Found at the URL: ftp://astro.estec.esa.nl/pub/XMM/documents/odf_icd.ps.gz.

[2] K. Galloway. XMM technical note: Periodic housekeeping telemetry definition. Technical Report XMM-SOC-TN-0040-SSD 0.2, ESA/SSD, August 5 1999.

[3] 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.

[4] EPIC System Team. EPCS user manual. Technical Report EPIC-EST-OP-002 Issue 2, Sepetember 1999.

[5] Giuseppe Vacanti. Interface specification for the xmm event attributes, April 1999. GV/XMM/1999-01.