Basic requirements for processing of EPIC science telemetry

L.Chiappetti - IFCTR
Issue 2 - 1 Feb 1997

1 Introduction

1.1 Purpose

The purpose of this technical note is to give a brief description of the kind of processing needed to make EPIC telemetry data readily usable by a scientific user. In fact, because of the need for an efficient usage of the downlink telemetry channel, and because of design constraints of EPIC sub-units, the actual telemetry format is quite complex, and is introducing significant layout changes with respect to the more natural photon list logical format. This logical format shall be recovered before presenting the data to the potential scientific user.

Such potential users may be members of the EPIC team, SOC Duty Scientists or XMM Guest Observers. As such the intended audience of this document may include persons in charge of developing software which handles EPIC data, either within the EPIC Consortium (e.g. EGSE team, Calibration Data Analysis Team) or within ESA (e.g. for ODF production).
[The information contained in this issue 2.0 of the document makes reference to the latest official information available, including updates after the EIDR documentation package (particularly for the interface documents); in particular reference documents are listed in section 5. ***In some cases it may also anticipate information not yet formalized.***

Items which are TBV, unclear, or subject to later internal revision (done by h/w people in red, to be done by CDAWG in blue) are identified in this draft with a bar on the margin and with boldface italics, like this paragraph.
A single bar indicates changes since issue 1.0.

1.2 General considerations

This document attempts to give a comprehensive view of the EPIC MOS and pn science data, how it looks like in the telemetry, and how it should look like to a Guest Observer (or other scientific user). When punctual information is needed, proper references to the latest official (applicable) documents is given. Such documents shall be consulted for complete information.
For ERMS data, which are collected by the RTU and packetized by the OBDH, information is provided in a separate document [Ref. R1]

The data in the logical format proposed here for use by the scientific users is described in terms of "files". This is a strong recommendation towards using individual disk files (e.g. the ODF shall be seen as a collection of files).
In the description of the layout of the "files" some reference to the current terminology of FITS binary tables is made [Ref. 1], but this does *not imply at all* a recommendation towards or against the usage of the FITS format, although it represents a recommendation towards a simple tabular organization. Some considerations on the actual physical disk file format are presented in section 4.
In particular the layout of a tabular file is presented using the definitions described below in section 1.3.4.
The term "administrative info" refers to the information associated to a file (a set of parameters applying to the entire file) which in a FITS file constitutes the header keywords.

1.3 Definitions

1.3.1 Spatial coordinate systems

The following set of coordinates are defined respectively for the MOS and pn CCDs in the EMCS or EPCS (figures are reported in Annex).

1.3.1.1 MOS Node detector coordinates:

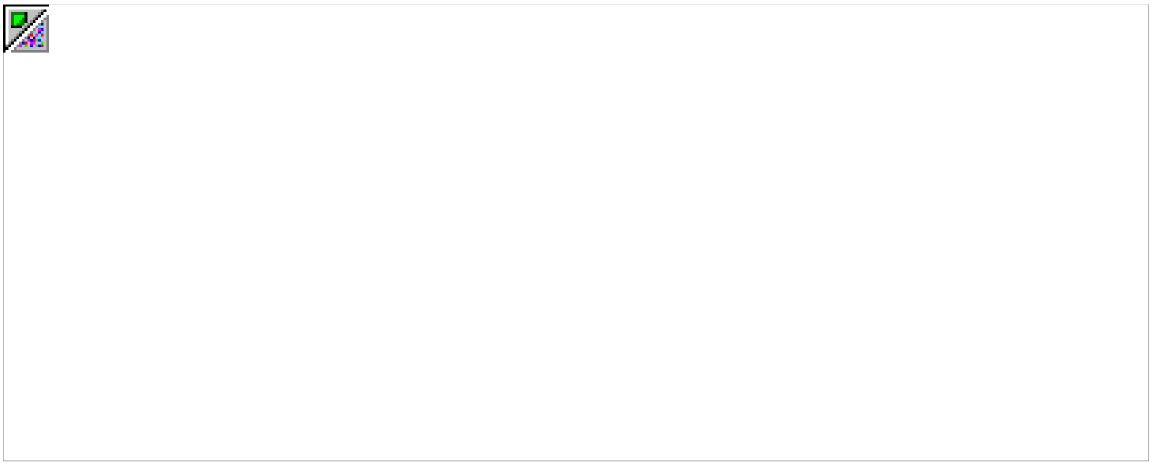
A set of coordinates having their origin in the current readout node. The readout nodes are shown in [Ref. M1 Fig. 3.1-2 as amended by addendum to M1 3.1.3, ***also M5 will be updated***].
The *prime readout node* is the one on the left hand side, and the *redundant readout node* is on the right hand side.
The node detector coordinates x_i,y_i are in pixel, range nominally 0 to 599, and are defined as follows:

for the prime readout node
x_i is 0 in the bottom left corner of the CCD (point indicated as 0,0 in [Ref. M1 Fig. 3.1-2]) and increases to the right up to a value of 599

for the redundant readout node
x_i is 0 in the bottom right corner of the CCD and increases to the left up to a value of 599

for both nodes
y_i is 0 in the bottom line of the CCD and increases to the top up to a value of 599

The (x_i,y_i) is an ideal system. The actual coordinate system used at EMCS level is the (u_i,v_i) system defined as follows (graph corresponds to the prime node case)



The v axis is identical to the y axis, but includes an extra region on top for overscan. This region is Δ_3 pixels high (Δ_3 ~ 2). The u axis is defined in the output register parallel to the x axis, but includes two extra regions on both sides for overscan. These regions are Δ_1 and Δ_2 pixels wide (Δ_1= Δ_2 ~ 5 each ? 10 total).

In addition the coordinates of an event falling in the central pixel of the 5∞5 cell used by the EMCR (x_c y_c) are returned as the coordinates of the top right |left (resp. for prime| redundant node) corner (u_0,v_0).
The (u_i,v_i) system is the coordinate system used throughout in the telemetry. It shall be corrected to the (x_i,y_i) system asap on the ground. Any (cosmic ray) event included in the overscan regions (i.e. outside of the 0-599 range of x_i or y_i) shall also be rejected asap.

The transformation is given by :

$x_i = u_i - 2 - \Delta_1$
$y_i = v_i - 2$

1.3.1.2 MOS Chip detector coordinates

A set of pixel coordinates X_i,Y_i having their origin in the prime readout node. Their relation with the node coordinates is the following :

$X_i = x_i(\text{prime}) \quad = 599-x_i(\text{redundant})$
$Y_i = y_i(\text{prime}) \quad = y_i(\text{redundant})$

These coordinates (as the node coordinates) are defined independently for each CCD.

1.3.1.3 MOS Camera-oriented chip detector coordinates

A set of pixel coordinates X_i,Y_i sharing a common orientation for each CCD in a given camera. Considering the focal plane layout and CCD numbering in [Ref. M1 Fig. 3.1-1 and amendment] one might assume an origin in the prime node of chip number 4. Therefore one has :

For chip 1	X_i=599-Y_i	Y_i =X_i
For chip 2	X_i=X_i	Y_i =Y_i
For chip 3	X_i=X_i	Y_i =Y_i
For chip 4	X_i=X_i	Y_i =Y_i
For chip 5	X_i=599-X_i	Y_i =599-Y_i
For chip 6	X_i=599-X_i	Y_i =599-Y_i
For chip 7	X_i=599-X_i	Y_i =599-Y_i

1.3.1.4 MOS Camera detector coordinates

A set of coordinates relevant to the entire focal plane has a natural origin in the centre of chip 1, and the same orientation as X_i,Y_i . One might define such coordinates in two ways :

- As a set of *conventional* pixel coordinates (useful to build a 1800∞1800 mosaic image of the entire focal plane) derived adding a couple of offsets ΔX,ΔY to X_i,Y_i as follows :

For chip 1	ΔX=-299	ΔY =-299
For chip 2	ΔX=+301	ΔY =-300
For chip 3	ΔX=-299	ΔY =-899
For chip 4	ΔX=-898	ΔY =-300
For chip 5	ΔX=-898	ΔY =+1
For chip 6	ΔX=-299	ΔY =+301
For chip 7	ΔX=+301	ΔY =+1

- As a set of *physical*coordinates (in mm or microns) derived from the conventional coordinates taking also into account the dead spacing used for CCD butting.

1.3.1.5 pn Chip detector coordinates

The numbering of the CCDs and quadrants is presented in [Addendum to Ref. P1, item 3] :

The four quadrants are numbered 0 to 3 in clockwise direction from the top left quadrant, with the CCDs seen from the backside (entrance window).

In each quadrant the three CCDs are numbered 0 to 2 (this is the *CCD id* in the data format [Ref. P3 5.5.3.1]) with Chip 0 being the one closest to the centre, and Chip 2 the farthest from it.
The overall CCD numbering (1 to 12) used in [Ref. P1 Fig. 3.1-1] is now obsolete from the point of view of on-board data, but ***may remain convenient for final scientific users on the ground***. There is the following correspondence table :

Quad 0 Chip 2 = #1	Quad 0 Chip 1 = #2	Quad 0 Chip 0 = #3	Quad 1 Chip 0 = #4	Quad 1 Chip 1 = #5	Quad 1 Chip 2 = #6
Quad 3 Chip 2 = #7	Quad 3 Chip 1 = #8	Quad 3 Chip 0 = #9	Quad 2 Chip 0 = #10	Quad 2 Chip 1 = #11	Quad 2 Chip 2 = #12

The chip detector coordinates x_i,y_i (or channel,line or column,line) are in pixel, range nominally 0 to 63 and 0 to 199, and are defined as follows:

for any chip irrespective of the quadrant
x_i is 0 in the right hand corner of the CCD looking with the CAMEX towards the bottom and increases to the left up to a value of 63

y_i is 0 in the line of the CCD closest to the CAMEX and increases farther from it to a value of 199

<u>This is the coordinates system used throughout in the telemetry.</u>

Although inconvenient for the scientific user, the fact that this reference systems has an orientation opposite to the usual Cartesian systems (and screen coordinate systems) is a matter of convenience for the electronics. Otherwise one might consider a flipped chip coordinate systems, flipping x_i to get the usual orientation).

1.3.1.6 pn Quadrant detector coordinates

Since the pn chips are relatively small (in terms of number of pixels), and the data are usually transmitted by quadrant, one might define a set of pixel coordinates X_i,Y_i relevant to each quadrant, and ranging respectively from 0 to 191 and 0 to 199.

The first obvious definition is to define $Y_i=y_i$ while one has $X_i=x_i+\Delta x$, where Δx depends on the chip and quadrant as follows :

0	+64	+128	0	+64	+128
+128	+64	0	+128	+64	0

This preserves the same (inconvenient) orientation of the chip detector coordinates, and again one might consider to flip the resulting X axis.

1.3.1.7 pn Camera-oriented coordinates

A set of pixel coordinates x_i,y_i or X_i,Y_i sharing a common orientation for each CCD or quadrant in a given camera. The more natural system (considering the figure quoted in 1.3.1.5 i.e. the view of the pn camera backside) is one having the origin in the bottom left corner of each chip or quadrant, and the standard Cartesian orientation, that is :

Camera-oriented chip coordinates

$x_i=x_i$ $y_i=199-y_i$	$x_i=x_i$ $y_i=199-y_i$	$x_i=x_i$ $y_i=199-y_i$	$x_i=x_i$ $y_i=199-y_i$	$x_i=x_i$ $y_i=199-y_i$	$x_i=x_i$ $y_i=199-y_i$
$x_i=63-x_i$ $y_i=y_i$	$x_i=63-x_i$ $y_i=y_i$	$x_i=63-x_i$ $y_i=y_i$	$x_i=63-x_i$ $y_i=y_i$	$x_i=63-x_i$ $y_i=y_i$	$x_i=63-x_i$ $y_i=y_i$

Camera-oriented quadrant coordinates

$X_i=X_i$ $Y_i=199-Y_i$	$X_i=X_i$ $Y_i=199-Y_i$
$X_i=191-X_i$ $Y_i=Y_i$	$X_i=191-X_i$ $Y_i=Y_i$

1.3.1.8 pn Camera detector coordinates

A set of coordinates relevant to the entire focal plane has naturally the same orientation as X_i,Y_i while its origin can either be in the detector centre (for consistency with the definition given in 1.3.1.4 for the MOS) or in the lower left corner of quadrant 3 (the latter is easier if one want to make a display of the entire focal plane). One might define such coordinates in two further ways , a conventional and a physical sense.

- As a set of *conventional* pixel coordinates (useful to build a 384∞400 mosaic image of the entire focal plane) derived adding a couple of offsets $\Delta X,\Delta Y$ to X_i,Y_i as follows :

	origin in the centre		origin in lower left corner	
For quad 0	$\Delta X=-192$	$\Delta Y = 0$	$\Delta X= 0$	$\Delta Y =+200$
For quad 1	$\Delta X= 0$	$\Delta Y = 0$	$\Delta X=+192$	$\Delta Y =+200$
For quad 2	$\Delta X= 0$	$\Delta Y =-200$	$\Delta X=+192$	$\Delta Y = 0$
For quad 3	$\Delta X=-192$	$\Delta Y =-200$	$\Delta X= 0$	$\Delta Y = 0$

- As a set of *physical* coordinates (in mm or microns) derived from the conventional coordinates taking also into account the fact that the size of the pixel adjacent to quadrant boundaries is larger (although there is no *dead* space).

1.3.1.9 Spacecraft coordinates

These shall be intended as *physical* coordinates (in mm or microns), and can be interpreted as :

- the physical camera detector coordinates described above
- the physical camera detector coordinates translated to have the origin in the intersection of the telescope optical axis with the focal plane.

- the above translated system, rotated to align the axes with a common TBD orientation for the three EPIC cameras (e.g. the spacecraft axes lying in the focal plane).

1.3.1.10 Celestial coordinates

The celestial coordinates α,δ of any pixel on any chip can be derived converting the pixel coordinates to the third form of spacecraft coordinates, and taking into account the nominal or actual satellite attitude and the eventual misalignments between telescope and star tracker axes.

The generation of spacecraft and celestial coordinates is *by no means* intended as an EPIC task, but should be care of ESA.

1.3.2 Energy coordinates

The energy of a photon is measured digitizing the charge deposited on a 12-bit scale. The conversion from ADC channels (hereafter just *channels*) to physical energy units (energy calibration) is *outside of the scope* of the present document. Hereafter the symbol E is reserved to physical energy in keV, while the symbol Q is used for energy/charge in digital channels.

In addition it must be remembered that a photon may deposit charge in more than one pixel (*splitting*), and that additional processes (from cosmic rays, to noise, to pile-up of more than one photon in the same or neighbouring pixels) may deposit charge.
In particular actions must be taken to reconstruct the charge content related to an event after splitting. This operation

is done partially on-board for the MOS camera (by the EMCR which may provide more than one "energy info" per event) and must be completed on the ground combining the appropriate info

is not done on-board by the EPIC pn camera, and must be done on the ground

1.3.3 Time coordinates

The arrival time of a photon cannot usually be tagged to an accuracy greater than the frame/cycle duration (frame/cycle time tagging is defined in later sections). There are however some specially designed modes which allow the reconstruction of a time with a better accuracy using additional information. This reconstruction is described in this note.

The present document does not specify time units (e.g. spacecraft time, µs, ms etc.) which may depend on the stage of data processing. However of course any algebraic operations among times implies the same consistent units are used.

It shall be noted that the times in EPIC science packets are usually times generated within EMCR/EPEA (and not within EMDH/EPDH unless explicitly stated). The following relationships hold between EMCR/EPEA, EMDH/EPDH and spacecraft times (Silvestri dixit to Balasini 29 Aug 95):

- EMCR and EPEA synchronize their coarse time (1 s resolution) every second with their EMDH or EPDH (when these receive the 1 Hz broadcast pulse from OBDH to get synchronized with the spacecraft clock).
- The fine time is generated within EMCR and EPEA and reset to zero at the same time of the ExDH 1 Hz synchronization of the coarse time.
- EMCR/EPEA coarse+fine time, although apparently stored in a 32-bit word, are actually 15+15 bits for MOS and 15+16 for pn, i.e. both shorter than the spacecraft time, and with a different format and resolution of LSB between each other, namely
  - in both cases the coarse time T_C has an LSB of 1 sec
  - in the MOS case the coarse time T_C uses 15 bits in a 16 bit word whose MSB is 0, therefore it can be used as normal signed positive integer.
  - in the MOS case the fine time T_F is 15 bit long (in a 16 bit word), and has an LSB of Δt=40 µs (this means that, apart from clock drifts, its actual range is 0-24999) therefore an elapsed time in second is obtained as $T=T_C + T_F \Delta t$
  - in the pn case the coarse time T_C uses 15 bits in a 16 bit word whose MSB is 1, and must therefore be extracted from it masking the MSB.
  - in the pn case the fine time T_F is 16 bit long, and unsigned, and has an LSB of Δt=16 µs, (this means that, apart from clock drifts, its actual range is 0-62499) therefore an elapsed time in second is obtained as as $T=T_C + T_F \Delta t$

- in the above formulae one can suggeest that if T is implemented as a double precision floating point value it shall preserve all the necessary precision
- To prevent ambiguities in time the EMCR/EPEA times are *reset* (restart from zero) by their EMDH/EPDH [Ref. M1/P1 URS-7.2.12.6] automatically at the start of each exposure. The absolute spacecraft time of this reset is recorded in the HK telemetry.
- In the event of long exposures such reset is also automatically done after 9 hours (programmable), which is less than the 9.1 hours after which a 32-bit time will recycle.
- One can think of the spacecraft clock as a clock keeping the absolute time, and of the EMCR/EPEA time as a stop watch that starts counting at the beginning of each exposure.

The correlation of the times in the EPIC packets with the spacecraft time, and ultimately with the Universal Time is is *by no means* intended as an EPIC task, but should be care of ESA.

1.3.4 Tabular file layout

In the description of a tabular file, the following terms, mutuated from the FITS `BINTABLE` terminology [Ref. 1], are used to define the fields (aka columns) in a record (aka row):

- Column — is a sequence number of the field. It has to intended for convenience of the description only. It does not imply a recommendation to use the physical order described here (the order shall be dictated by programming convenience, i.e. word alignment, although it is desirable that the same logical information occurs in the same order in different files).
- TTYPE — is the name of the field. The final names have to be agreed. The names used here are just for convenience to make clear when the same information is referred to.
- TUNIT — are the physical units in which the quantity is expressed (if known)
- TFORM — is the "minimum" format of the quantity in the FITS notation `n`*X*. `n` indicates the dimensionality (usually 1, but a quantity may also be present as an array of n elements), while *X* is coded as `I` for 16-bit integers, `J` for 32-bit integers, and `E` for 32-bit floating point. Since this document (outside of section 4) does not mandate for physical data format, this has not to be intended as a mandatory requirement (in particular it does not imply the standard IEEE representation and endianness). Moreover, indicating a quantity (e.g. an energy field, originally of 12 or less bits) as `1I`, it is not mandated to use 16-bit integers. It just says that such a quantity, if one desires to use only "standard" data types (i.e. 16- 32- and 64-byte ones) must be padded at least into a 16-bit integer. If other convenience reasons for instance suggest the usage of 32-bit integers everywhere, this is possible.
- Content — finally describes what the field is, in terms of data present in the telemetry, and of the reprocessing described.

1.3.5 Symbols used

C_f	count info
δy_i	differential y coordinate (in compressed timing mode)
d_i	generic data value
e_i	a generic event (e may be any lowercase letter)
E	energy (in keV)
f	generic frame index (f may assume any lowercase value)
H_f	frame header
i	generic event index (i may be any numeric value 1...)
I_i	event pattern identifier
K_f	reduced frame header
N_c,N_r	number of columns,rows used in offset/threshold computation
O^c,O^r	arrays of column or row offsets (MOS)
o_i	pixel offset values (pn)
p_i	generic value of pixel content
Q	energy or charge (in ADC channels)
s_i	pixel status values (pn)
t_i	event time
t_f	frame time (or cycle time)
T_f	frame trailer or time info
x_i,y_i	MOS node or pn chip detector coordinates (see 1.3.1.1/5)
X_i,Y_i	MOS chip or pn quadrant detector coordinates (see 1.3.1.2/6)
x_i,y_i	camera-oriented pn chip detector coordinates (see 1.3.1.7)
X_i,Y_i	camera-oriented MOS chip or pn quadrant detector coordinates (see 1.3.1.3/7)

1.3.6 Acronyms

Only acronyms specific of this document, or otherwise of uncommon use will be listed here. For the usual EPIC or ESA acronyms (EMCS, EPCS, etc. EDU, APID, etc.) see the reference documents.

ASF	Ancillary Science File
CDAWG	Calibration Data Analysis Working Group
ELF	Event List File
HKF	HouseKeeping File
FIF	Frame Image File
FITS	Flexible Image Transport Systems
NDC	Non-Destructive data Compression (in EMDH)
ODF	Observation Data file (see [Ref.4])
OVF	Offset/Variance File
RELF	Raw Event List File
SDPM	Science Data Processing Mode (in EMCR)

2 Science data

Note that now science data packets (Science Telemetry Reports) are differentiated using both the SID, and the packet subtype.

2.1 MOS Imaging packets

These packets are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of 1
a SID of **1** [Ref.M5 5.10.2.12]

There is a separate stream of Imaging packets for *each* CCD (or node in case of hypothetic 2-node operation, as derived from [Ref. M2 URS-7.2.5.6.1.3]) operated in an imaging readout option (either full field of view frame store from 1 node, from 2 nodes, window option or refreshed frame store [Ref. M4 section 2.3.1]) in Prime or Fast Mode [Ref. M1 sections 3.4.2/3]. It is assumed that the EDU associated to the CCD or node is operating in Imaging SDPM [Ref. M3 section 6.2.2.].

2.1.1 Telemetry format

For a complete reference see [Ref. M5 section 5.10.2.12/13].
For what concerns here it has to be noted that the original stream of photons falling on the given CCD with the following characteristics :

An event e_i at time t_i with energy E_i at CCD pixel position X_i,Y_i

is detected during a given frame. In what follows events (and associated data formats) in a different frame will be identified with a different lowercase letter, e.g. a_i,b_i,c_i ... with the index i running continuously from 1 onwards. The same letter a,b,c... is used to index items related to the specific frame (where f might be used to index a generic frame).

An event e_i is in general *one* X-ray *photon* (as reconstructed by the EDU), but might also correspond to a *pixel hit by a cosmic ray*, or to a *pile-up of more events* (in particular geometric pile-ups of two photons adjacent in diagonal direction will be characterized by a precise set of pattern ids), or to an *incomplete reconstruction*.

Event data from each frame are included between a frame header H_f and a frame trailer T_f. Data follow without solution of continuity from one packet to the next. In the following example we assume that frames a,b,c,d are completely contained in packet 1, frame e is split across packet 1 and 2 and frames f and following continue from packet 2 onwards.
The resulting data stream will be :

Packet 1: $H_a\ a_1\ a_2 ... a_n\ T_a H_b\ b_{n+1}\ b_{n+2} ... b_m\ T_b\ H_c\ c_{m+1}$ ... ... $d_p\ T_d H_e\ e_{p+1} .. e_q$
Packet 2: $K_e\ e_{q+1} ... e_r\ T_e H_f\ f_{r+1}$ ... ...

The item K_f (which is located at the beginning of the data field proper of a packet instead of a full header H_f only when frame f is split across the previous and the current packet) is a reduced header.

About the content of the 96-bit event format e_i [Ref. M5 quoted] one shall note that :

- There is no event time t_i in it. The only timing information is the frame time t_f contained in the frame trailer T_f.
- the coordinates x_i,y_i are *node detector coordinates* u_i,v_i
- the energy information is in the form of a digitized charge (in channels) as a set of four values Q_{1i},Q_{2i},Q_{3i},Q_{4i} (defined in Ref. [M5 pag. 32] ,we use here the notation $Q_{1..4}$ instead of $E_{1..4}$ as in [M1,M2,M3,M5] to make clear it is not the final energy.
- there is a pattern identifier I_i generated from the EDU (according to current - programmable - pattern library, for example see [Ref. M1 pag. C-2])
- there is pixel counter P_i of the pixels above threshold in the guard ring

About the content of the headers and trailers one shall note that they contain information which either :

- is used to identify the packet (for sorting) and is the same for all frames in a given packet (the combination of CCD id and node id)
- is used to identify the frame and varies monotonically from one frame to the next (the frame number)
- is used to time tag the frame and varies monotonically from one frame to the next (the frame time t_f (implemented as a couple T_C,T_F, see 1.3.3. above) defined as the end of the frame (-0 $+\Delta_{trans}$ ***TBC*** where Δ_{trans} is the frame transfer time)
- are of auxiliary nature and may vary randomly from one frame to the next (the various *event* and *pixel counters* and the *FIFO overflow flag* allpresent in the frame trailer, or the *Gatti flag* in the frame header)
- are of auxiliary nature and should not vary from one frame to the next within the same exposure (they record commanded parameters, like the *EDU mode* and *threshold*, *Frame Integration Time, window parameters e* and *EMDH lower and upper thresholds*. in the trailer)

2.1.2 Reprocessing

The primary purpose of the reprocessing is to regenerate the original event stream in the readily usable form of an event list, with its scientifically usable information i.e. the best estimate of time t_i , of energy E_i and the CCD pixel position X_i,Y_i (TBC for flight processing) or X_i,Y_i (the latter is the choice for EGSE).
This purpose is fulfilled by the generation of a reformatted event list file (ELF).

As a secondary purpose it should be considered the production of an ancillary science file (ASF) containing the frame-related information which does not fit the reformatted event list.

It is suggested to create a set of separate files (event list file and ancillary science file) per CCD.

The following is the suggested reprocessing :

- as a prerequisite only the relevant packets shall be processed, extracting them from the overall stream using APID, packet type, subtype and SID.
- the *CCD id* and *node id* in the first *frame header* or *reduced frame header* in each packet shall be tested, and the events contained in the packet shall be attributed to the relevant event list file according to the CCD id. The CCD id shall be saved in the event file list administrative info.
- the commandable parameters (*EDU mode* and *threshold*, *Frame Integration Time*, *window parameters* and *EMDH lower and upper threshold*) in the first frame header (or trailer) of the first packet of a given CCD stream shall be saved in the event list file administrative info.
- the value of such parameters in subsequent packet shall be checked not to change. In the case of a change the current event list file shall be closed, and a new one opened (this is assumed to happen only at exposure boundaries).
- the node coordinates x_i,y_i of each event shall be converted (TBC for flight) to camera-oriented chip detector coordinates X_i,Y_i while for EGSE purposes to chip detector coordinates X_i,Y_i); the conversion to camera detector coordinates, or to celestial coordinates should be considered in a later processing and not at this stage.
- each event shall be assigned an event time equal to its frame time ($t_i =_{df} T_f$) using 1.3.3.
- the array of event energy/charge info $Q_{1i},Q_{2i},Q_{3i},Q_{4i}$ shall be complemented reconstructing
  - the total event charge $Q_{Ti}=Q_{1i}+Q_{2i}$
  - the inner background estimate $Q_{B3i}=Q_{3i} / (9-N_i)$
  - the outer background estimate $Q_{B4i}=Q_{4i} / (16-N_i)$

  and the seven values $Q_{Ti},Q_{B3i},Q_{B4i},Q_{1i},Q_{2i},Q_{3i},Q_{4i}$ saved to the event list file, together with the peripheral pixel counter N_i
- the event pattern id I_i must be saved to the event list file.
  - events with a given subset of "bad" pattern id's (e.g. "extended" ones) are not filtered out at this stage (they are anyhow written to the event list file).
  - events corresponding to "diagonal geometric pile-up" (i.e. I_i = P3,P5,P7,P9 in [Ref. M1 pag. C-2]) are reconstructed at this stage (this is the only alteration of the original stream at this stage). This implies that for each "diagonal" event, two events are written to the event list file according to the following rules :

    one event with X_i,Y_i or X_i,Y_i as derived usually from x_i,y_i of the central pixel
    $Q_{Ti}=Q_{1i}$

    another event with X_i,Y_i or X_i,Y_i s derived from $x_i\pm1,y_i\pm1$ (the coordinate of the "other" black pixel in the figure in [Ref. M1 pag. C-2]
    $Q_{Ti}=Q_{12}$

    for both events Q_{B3i},Q_{B4i}, and all other parameters are an identical copy.

- the auxiliary frame counters are saved to the ancillary science file

2.1.3 Output file format

2.1.3.1 Event list file

A tabular file with one row per event (or reconstructed event). The following is the event format according to the conventions in sections 1.3.4 and 4.

Column	TTYPE	TUNIT	TFORM	Content
1	`TIME`	`s`	1D	frame time t_i
2	`X`	`PIXEL`	1I	CCD coordinate X_i or X_i
3	`Y`	`PIXEL`	1I	CCD coordinate Y_i or Y_i
4	`ENERGY`	`CHANNEL`	7I	Q_{Ti},Q_{B3i},Q_{B4i},$Q_{1..4i}$
5	`PATTERN`	`--`	1I	pattern id I_i
6	`PERIPIX`	`PIXEL`	1I	periph. pixel counter N_i

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EMCS1' or 'EMCS2'
CHIP     = CCD id (1-7)
NODE     = 'Prime' or 'Redundant'
DATATYPE = 'IMAGING'
EDU      = EDU number (from frame trailer, redundant)
EDUMODE  = value of the EDU mode (redundant ?)
EDUTHRES = value of the EDU threshold
FRMTIME  = value of the frame integration time
WINDOWX0 = window X0  (see note)
WINDOWY0 = window Y0  (see note)
WINDOWDX = window ΔX   (see note)
WINDOWDY = window ΔY  (see note)
EMDHLOW  = value of the EMDH lower threshold
EMDHHIGH = value of the EMDH upper threshold
```

The window parameters shall be converted in such a way that they locate correctly the window in the <u>same</u> coordinate systems used for events in the event list file.

Additional administrative info related to the EMCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers.

2.1.3.2 Ancillary science file

A tabular file with one row per frame with the following format according to the conventions in in sections 1.3.4 and 4

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	frame time T_i
2	FRAMENO	frame number	1I	frame counter
3	NPIXEL	pixel	1J	pixel count
4	NVALID	event	1I	valid event counter
5	NBELOW	event	1I	events below low threshold
6	NABOVE	event	1I	events above high threshold
7	FIFOOVER	TBD	1I	FIFO overflow flag

The administrative information shall be the same as described in 2.1.3.1.

2.2 MOS Imaging reduced packets

These packets (called improperly Imaging compressed packets in Ref. [M1]) are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of 1
a SID of **2** [Ref.M5 5.10.2.12]

There will be a stream of Imaging reduced packets for *such* CCD (or node in case of hypothetic 2-node operation, as derived from [Ref. M2 URS-7.2.5.6.1.3]) operated in an imaging readout option (see 2.1 above) in Prime or Fast Mode , *in the case* that the EDU associated to the CCD or node is operating in Threshold SDPM [Ref. M3 section 6.2.2.].
The EDU Threshold SDPM will be operated on command in case of increased CTI.

2.2.1 Telemetry format

For a complete reference see [Ref. M5 section 5.10.2.12/13].
About the way the original stream of photons is inserted in the telemetry, the same considerations given above in section 2.1.1 hold, with the following differences :

- The event format e_i is 48-bit long.
- the energy information is in the form of a single value corresponding to the total event charge $Q_{Ti}=Q_{1i}+Q_{2i}$ or $Q_{1i}+0$ according to the HBR setting as described in [Ref. M5 5.10.2.13.1].

2.2.2 Reprocessing

The same considerations given above in section 2.1.2 hold, with the following exceptions :

- the total event charge Q_{Ti} shall be inserted directly in the event list file
- the peripheral pixel counter shall be inserted directly in the event list file
- the reconstruction of diagonal pile-up events is no longer possible

2.2.3 Output file format

2.2.3.1 Event list file

A tabular file with one row per event. The following is the event format according to the conventions in sections 1.3.4 and 4.

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	frame time t_i
2	X	PIXEL	1I	CCD coordinate X_i or X_i
3	Y	PIXEL	1I	CCD coordinate Y_i or Y_i
4	ENERGY	CHANNEL	1I	Q_{Ti}
5	PATTERN	--	1I	pattern id I_i
6	PERIPIX	number	1I	periph. pixel counter N_i

The same administrative info listed in section 2.1.3.1 shall also be present (but `DATATYPE = 'IMAGING REDUCED'`).

Post-facto event reconstruction (combining adjacent monopixel events which should belong to a single photon) is TBD.

2.2.3.2 Ancillary science file

The same as described in section 2.1.3.2.

2.3 MOS Timing packets

These packets are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of **2**
a SID of **3** [Ref.M5 5.10.2.12]

There is a stream of Timing packets for *that* CCD (or node in case of hypothetic 2-node operation, as derived from [Ref. M2 URS-7.2.5.6.1.3]) operated in timing readout option [Ref. M4 section 2.3.2]) in Fast Mode [Ref. M1 section 3.4.3, this will usually be the central CCD but could be redefined in case of failures], in the case the EMDH is **not** operating the Non-Destructive Compression (NDC) option [Ref. M2 section 7.2.3.6]. It is assumed that the EDU associated to the CCD or node is operating in Timing SDPM [Ref. M3 section 6.2.2.].

2.3.1 Telemetry format

For a complete reference see [Ref. M1 Appendix E & F, Ref. M2 7.2.5.6.2 and Appendix A.2].
For what concerns here, given the general definitions in section 2.1.1 above, the following considerations hold about the the way the original stream of photons is inserted in the telemetry :

An event e_i at time t_i with energy E_i at CCD pixel position X_i,Y_i

is detected during a given cycle. The cycle is defined in [Ref.M1 E.3.7.2.7 andM5 pag 26]. In the following a cycle will be identified with the same notation used for frames in 2.1.1.
A time-info format T_f is generated at the cadence of each cycle. A time-info has the same format as a trailer, with the exception of two flag bits [Ref. M5 pag. 58].

A proper "frame "header H_a and "frame" trailer T_∞ are present only at the beginning and end of an exposure (the trailer T_∞ is necessary because the last cycle may not be completed yet).
Data follow without solution of continuity from one packet to the next, interleaving event formats and time-info formats.

The resulting data stream will be :

Packet 1: $\mathbf{H_a}$ a_1 a_2 ...a_n $T_a$$b_{n+1}$ b_{n+2} ...b_m T_b c_{m+1} ... ... d_p $T_d$$e_{p+1}$.. e_q
Packet 2: K_e e_{q+1} ...e_r T_e f_{r+1} ... ...
...
Packet N: K_x x_{u+1} ... ... y_w T_y z_{w+1} ... z_x $\mathbf{T_\infty}$

All packets but the first start always with a reduced header K_f which may differ from any other of the same kind only for the presence of the cycle number.

About the content of the 48-bit event format e_i [Ref. M5 quoted] one shall note that :

- Its layout is apparently the same of the imaging reduced event format
- There is no event time t_i in it. However accurate timing information can be derived combining the cycle time t_f contained in the time info T_f with the y_i event coordinate.
- the coordinate x_i is in *node detector coordinates*

- the coordinate y_i is not a spatial coordinate, but it is a counter increasing at each integration of a window in the vertical direction, corresponding to a time bin
- the energy information is in the form of a digitized charge content of a pixel (in channels) as single values Q_i
- there is a pattern identifier I_i generated from the EDU (which in this case just resolve single from double events [Ref. M1 pag. C-3]. Note that inclusion of double events (patterns 1 and 2) in the telemetry can be disabled at EMDH level [Ref. M2 7.2.3.5]. Also note pattern numbers 3 and 4 are not used in flight.

About the content of the single header and trailer and of time info (which are equal to the trailer but for a flag bit), the general considerations given in 2.1.1 hold (the main differences are the presence of the cycle counter instead of the frame counter, and the meaning of the upper thresholding counter).

2.3.2 Reprocessing

The primary purpose of the reprocessing is to regenerate the original event stream in the readily usable form of an event list, with its scientifically usable information i.e. the best estimate of time t_i , of energy E_i and the CCD pixel position X_i (TBC for flight processing) or X_i (the latter is the choice for EGSE).
It has to be noted that in this mode spatial information is present in one dimension only.
This purpose is fulfilled by the generation of a reformatted event list file (ELF).

As a secondary purpose it should be considered the production of an ancillary science file (ASF) containing the cycle-related information which does not fit the reformatted event list.

It is suggested to create a set of separate files (event list file and ancillary science file) per CCD.

The following is the suggested reprocessing :

- as a prerequisite only the relevant packets shall be processed, extracting them from the overall stream using APID, packet type, subtype and SID.
- the *CCD id* and *node id* in the first packet *"frame" header* or *reduced header* in each other packet shall be tested, and the events contained in the packet shall be attributed to the relevant event list file according to the CCD id. The CCD id shall be saved in the event file list administrative info.
- the commandable parameters in the first packet "frame" header (or time info or trailer) of a given CCD stream shall be saved in the event list file administrative info.
- the node coordinates x_i of each event shall be converted (TBC for flight) to camera-oriented chip detector coordinates X_i while for EGSE purposes to chip detector coordinates X_i); the conversion to camera detector coordinates, or to celestial coordinates should be considered in a later processing and not at this stage
- each event shall be assigned an event time $t_i=T_{start}+T_W\, y_i$ where
  T_{start} is the start of the cycle when y_i is assumed to be zero
  T_W is the time necessary to expose a window, and stack it into one row (and depends on the sizes of the window).
  the addition shall of course take place with all times in the same units (see e.g. 1.3.3)
  the relation with the cycle time in telemetry is $T_{start} = T_f$ (-0 + Δ_L ***TBC*** where Δ_L is 1023 or 1024 TBC ∞ the line time)
- nevertheless the raw values of the cycle time T_f (or T_∞) and of the y_i coordinate shall be saved to an event list file too (see below for details).
- the event energy/charge and pattern id information are transferred to the event list file since the reconstruction of $Q_i=Q_{i1}+Q_{i2}$ in case of double events has already been done on board. The pattern id can just be used for later data selection.

∑ the auxiliary frame counters are saved to the ancillary science file

2.3.3 Output file format

2.3.3.1 Event list file

There are in principle two possibilities for the event list file. One is to produce a single event list file which contains both raw information and reconstructed information. The other one is to produce first a raw event list file name (RELF) (with minimum reformatting), and then process it to produce a final event list file. In the case of ODF production both files (if produced) shall be given to the Guest Observer.

All files are tabular files with one row per event (or reconstructed event). The following are the event formats according to the conventions in sections 1.3.4 and 4.

In the case a single event list file is produced

Column	TTYPE	TUNIT	TFORM	Content
a	TIME	s	1D	reconstructed time t_i
b	X	PIXEL	1I	CCD coordinate $X_i$or X_i
c	ENERGY	CHANNEL	1I	Q_i
d	PATTERN	--	1I	pattern id I_i
e	PERIPIX	number	1I	periph.pixels above thresh.
f	RAWTIME	s	1D	raw cycle time T_f
g	RAWY	PIXEL	1I	raw node coordinate y_i

If a single event list file is produced, it shall contain all columns a,b,c,d,e,f,g. If two files are produced, the RELF shall contain (in the order given) columns f,b,g,c,d,e and the final ELF instead columns a,b,c,d,e only.
It could be advisable to preserve the raw y_i (column g) info also in the final ELF, since the most accurate estimate of the event time depends on a fine correction based on y_i-Y_* where Y_* is the best estimate of the X-ray source position (known to the final user, which may want to perform this fine correction).
The case of two separate files (RELF and ELF) is preferred since double event reconstruction has to be done on the ground (i.e. the two files do not contain the same number of events).

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EMCS1' or 'EMCS2'
CHIP     = CCD id (1-7)
NODE     = 'Prime' or 'Redundant'
DATATYPE = 'TIMING' or 'TIMING RAW'|'TIMING FINAL'
EDU      = EDU number (from frame trailer, redundant)
EDUMODE  = value of the EDU mode (redundant ?)
EDUTHRES = value of the EDU threshold
FRMTIME  = value of the frame integration time
WINDOWX0 = window X0 (see note)
WINDOWY0 = window Y0 (see note)
WINDOWDX = window ΔX  (see note)
WINDOWDY = window ΔY  (see note)
EMDHLOW  = value of the EMDH lower threshold
EMDHHIGH = value of the EMDH upper threshold
```

The window parameters (of which Y_0 does not apply to the timing mode while ΔY may apply **if it is not hardcoded in the sequencer TBC**) shall be converted in such a way that they locate correctly the window in the same coordinate systems used for events in the event list file

Additional administrative info related to the EMCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers (e.g. window position and size).

2.3.3.2 Ancillary science file

A tabular file with one row per frame with the following format according to the conventions in in sections 1.3.4 and 4

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	cycle time T_i or T_∞
2	CYCLENO	cycle number	1I	cycle counter
3	NPIXEL	pixel	1J	pixel count
4	NVALID	event	1I	valid event counter
5	NBELOW	event	1I	events below low threshold
6	NABOVE	event	1I	events above high threshold
7	FIFOOVER	TBD	1I	FIFO overflow flag

NABOVE will count also events discriminated by pattern discrimination if enabled.
The administrative information shall be the same as described in 2.3.3.1.

2.4 MOS Timing compressed packets

These packets are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of **3**
a SID of **4** [[Ref.M5 5.10.2.12]

There is a stream of Timing compressed packets for *that* CCD (or node) operated in timing readout option [Ref. M4 section 2.3.2]) in Fast Mode [Ref. M1 section 3.4.3, this will usually be the central CCD but could be redefined in case of failures], in the case the EMDH **is** operating the Non-Destructive Compression (NDC) option [Ref. M2 section 7.2.3.6]. It is assumed that the EDU associated to the CCD or node is operating in Timing SDPM [Ref. M3 section 6.2.2.].
The operation of NDC will be enabled on command to reduce the telemetry load for brighter sources

2.4.1 Telemetry format

For a complete reference see [Ref. M5 section 5.10.2.12/13].
The considerations about the the way the original stream of photons is inserted in the telemetry are the same presented in 2.3.1, with the only difference that the event formats in the stream :

Packet 1: $\mathbf{H_a}\ a_1\ \mathit{a_2} \ldots \mathit{a_n}\ T_a b_{n+1}\ \mathit{b_{n+2}} \ldots \mathit{b_m}\ T_b\ c_{m+1} \ldots \ldots \ldots \mathit{d_p}\ T_d e_{p+1} \ldots e_q$
Packet 2: $K_e\ \mathit{e_{q+1}} \ldots \mathit{e_r}\ T_e\ f_{r+1} \ldots \ldots \ldots$
...
Packet N: $K_x\ \mathit{x_{u+1}} \ldots \ldots \ldots \mathit{y_w}\ T_y\ z_{w+1} \ldots \mathit{z_x}\ \mathbf{T_\infty}$

are in a priori unpredictable sequence of uncompressed and compressed events (indicated by *italics* in the scheme above). The first event after a header or time-info shall always be uncompressed. Subsequent events are compressed when possible (i.e. when the difference δy_i in y-coordinate between two subsequent events is less or equal to 7).

It must be noted that a compressed event format contains only energy information Q_i and differential y-coordinate information δy_i (by which y_i and ultimately time can be derived). There is no spatial x-coordinate information, no pattern id and no peripheral pixel info.

2.4.2 Reprocessing

The reprocessing required is the same as described in section 2.3.2 above, with the provision that compressed events are recognised (by the flag bit in the relevant data word), and converted to uncompressed format as follows :

- the node coordinate x_i of a compressed events shall assume a dummy value (it is suggested to use an illegal value, like -1).
- the other coordinate y_i of a compressed event is reconstructed as $y_i = y_{i-1} + \delta y_i$
- the pattern id of a compressed event is undefined and shall be flagged with an "illegal" value

2.4.3 Output file format

2.4.3.1 Event list file

The same as described in section 2.3.3.1, noting however that :

- the values of the X coordinate assumes value -1 for events reconstructed from compressed events (these coordinates are meaningless and must no be used for analysis)
- the values of the PATTERN and PERIPIX assume value -1 for events reconstructed from compressed events (these values are meaningless and must no be used for analysis)

The administrative info is the same as described in 2.3.3.1, only if wished DATATYPE may assume the value 'TIMING COMPRESSED' as a reminder (although the file layout is the same).

2.4.3.2 Ancillary science file

The same as described in section 2.3.3.2

2.5 MOS CCD diagnostics packets

These packets are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of **3**
a SID of **5** [Ref.M5 5.10.2.12]

There is a stream of CCD Diagnostics packets for *one* CCD at any time, when the EDU associated to the CCD or node is operating in Transparent SDPM [Ref. M3 section 6.2.2.], and the EMDH is just packetising the received data, i.e. the camera is in CCD Diagnostics Mode [Ref. M1 section 3.5.7].
This mode will be used for checkout and diagnostics (in flight presumably during set-aside observations requested by SOC) , and its data will not be relevant for Guest Observers.

2.5.1 Telemetry format

For a complete reference see [Ref. M5 section 5.10.2.12/13].
In this mode the information transmitted is not directly related to detected photons but to the charge content of each pixel (including noise etc.).

There is a fixed (programmable) number N of pixels for a given activation ("shot") of the CCD Diagnostics Mode, the maximum value of N is 360000 (***or 3672200 TBC*** if overscan is not screened at EMCR level), i.e. an entire CCD frame (the "good" frame will be read after skipping a programmable number of "unwanted" frames). This mode can also be used in conjunction with window and timing readout.
Therefore there will be a fixed number of packets per shot M, as a function of N.

The N pixel values p_i follow without solution of continuity from one packet to the next, and the resulting data stream will be :

Packet 1: $\mathbf{H_0}$ p_1 p_2 ...... p_q
Packet 2: K_0 p_{q+1} ....p_r
...
Packet M: K_0 p_{u+1} ... ... p_N $\mathbf{T_\infty}$

with a single "frame "header H_0 and "frame" trailer T_∞ present only at the beginning and end of the shot, and an identical reduced header K_0 present at the beginning of any packet but the first.

The possibility to have one exposure in this mode comprising more shots is to be defined operationally.

2.5.2 Reprocessing

In the case the CCD diagnostics mode is used in conjunction with full imaging readout (or even window readout) the logical data structure produced is an image of the CCD charge content. The same (although possibly with a different interpretation) holds also if it is used in conjunction with timing readout.

The reprocessing shall therefore reconstruct the pixel stream $p_1...p_N$ in the form of an image (each "energy data format" as described in [Ref. M2 pag. A-7] shall be converted into a 16-bit charge value zeroing the first 4 bits).

The few useful information in the header H_0 and trailer T_∞ shall be put in the administrative info of the image file.
The pixel count in the trailer shall be used only for a (redundant ?) check of the value N, and also the reduced headers shall be used only for verifying the unpacking of the stream.

There is no ancillary science file associated to the header/trailer content.

The above assumes the baseline of one exposure is a single shot of one frame.
If an exposure can comprise more shots, one may consider chaining the images derived from each shot into a single file, and recording in an *ancillary science file* the times of each shot (from the relevant trailer).

2.5.3 Output file format

2.5.3.1 Frame image file

The frame of a single shot is a 2-d image with N pixels of 16-bit width.
The dimension of the image is $N_x \infty N_y = N$.
In the case of full imaging readout (with overscan already eliminated) one has $N_x = N_y = 600$, while in other cases N_x and N_y shall be derived from frame headers.

In the hypothetic case of multi-shot exposures the data can be concatenated as a sequence of "image extensions" or as a 3-d image with dimensions $N_{shot} \infty N_x \infty N_y$

At least the following administrative info shall also be present.

`SATELLIT`	=	`'XMM'`
`INSTRUME`	=	`'EPIC'`
`CHAIN`	=	`'EMCS1'` or `'EMCS2'`
`CHIP`	=	CCD id (1-7)
`NODE`	=	`'Prime'` or `'Redundant'`
`DATATYPE`	=	is unnecessary, implicit being this an image file
`FRMTIME`	=	value of the frame integration time
`REFTIME`	=	value of the time in the trailer (goes in ASF for ***multi-shot case***)

Additional administrative info related to the EMCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers (e.g. window position and size).

2.6 MOS Offset/Variance packets

These packets are characterised by :

an APID either of EMCS1 or EMCS2 [Ref. M5 5.10.2],
a Packet Type of 15
a Packet Subtype of **4**
a SID of **6** [Ref.M5 5.10.2.12]

There is a stream of Offset/Variance packets for *one* CCD at any time, when the EDU associated to the CCD or node is operating in Transparent SDPM [Ref. M3 section 6.2.2.], and the EMDH is performing the Offset/Variance Computation [Ref. M2 section 7.2.3.8] , i.e. the camera is in CCD Offset/Variance Mode [Ref. M1 section 3.5.6].
This mode is likely to be used regularly (as part of the observation setup procedure). Its data will be used by SOC to adjust or verify the offset and threshold setup. As such they may not be relevant for Guest Observers (but inclusion in ODF could be considered as an option).

2.6.1 Telemetry format

For a complete reference see [Ref. M5 section 5.10.2.12/13].
In this mode the information transmitted is not directly related to detected photons but to the result of a processing of the charge content of a selected subset of pixels (essentially for an assessment of the noise level).

The data generated on-board by the Offset/Variance Computation are :

an array of N_c column-averages $N^c(i)$
an array of N_r row averages $N^r(j)$
a scalar value (total average) A_{tot}
an array of N_c column offsets $O^c(i)$
an array of N_r row offsets $O^r(i)$
a scalar value (total variance) σ

This operation is made once for a given activation ("shot") of the Offset/Variance Mode, and N_c and N_r are identically equal to 600, i.e. an entire full imaging CCD frame is processed. For timing and window options one shall use Diagnostic mode and process such data on the ground.

The data transmitted to ground are the arrays O^c, O^r and the scalar σ in the trailer (***how can ont tell row or column in [M5]; documents [M1] and [M2] are not totally clear on the matter***), i.e. an array of $L=N_c+N_r=1201$ values (since the order in which they follow is not defined anywhere I will refer to this array as the data values $d_1...d_L$).
Therefore there will be a fixed number of packets per shot M, as a function of L.

The L pixel values d_i follow without solution of continuity from one packet to the next, and the resulting data stream will be :

Packet 1: $\mathbf{H_0}$ d_1 d_2 ...... d_q
Packet 2: K_0 d_{q+1} ....d_r
Packet M: K_0 d_{u+1} ... ... d_L $\mathbf{T_\infty}$

with a single "frame "header H_0 and "frame" trailer T_∞ present only at the beginning and end of the shot, and an identical reduced header K_0 present at the beginning of any packet but the first (headers and trailers are similar but not identical to the ones of the Diagnostic packets).

The possibility to have one exposure in this mode comprising more shots is to be defined operationally.

2.6.2 Reprocessing

The logical data structure produced by the Offset/Variance mode is the ensemble of the two offset arrays and of the scalar variance.

The reprocessing shall extract the values from the data stream $d_1...d_L$; the values shall be converted into a 16-bit charge value zeroing the first 4 bits and shall be parsed into the two arrays O^c, O^r ; the scalar σ is extracted from the trailer.

The extracted values shall be saved to an Offset/Variance File (OVF).
The generation of the file is desirable to save the data e.g. for background (light contamination) analysis, otherwise the extracted values might be passed directly to the analysis procedure (whose definition is outside of the scope of the present note).

The few useful information in the header H_0 and trailer T_∞ shall be put also in the file (most likely in the administrative info).
The pixel count in the trailer shall be used only for a (redundant ?) check of the value L, and also the reduced headers shall be used only for verifying the unpacking of the stream.

There is no ancillary science file associated to the header/trailer content.

The above assumes the baseline of one exposure is a single shot of one frame.
If an exposure can comprise more shots, the simplest approach is to have each shot to correspond to an individual record in the OVF.

2.6.3 Output file format

2.6.3.1 Offset/Variance file

A tabular file with one row per shot (usually there will be a single shot). The following are the shot formats according to the conventions in sections 1.3.4 and 4.

Column	TTYPE	TUNIT	TFORM	Content
1	`TIME`	`s`	1D	time in the trailer t_∞
2	`ROWOFFST`	`CHANNEL`	nI	the array O^r, $n=N_r$
3	`COLOFFST`	`CHANNEL`	mI	the array O^c, $m=N_c$
4	`VARIANCE`	`CHANNEL`	1I	the scalar σ

In the case the file consists ***always*** of a single shot, one might dispense with the `TIME` and `VARIANCE` columns and place the value in the administrative info.

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EMCS1' or 'EMCS2'
CHIP     = CCD id (1-7)
NODE     = 'Prime' or 'Redundant'
DATATYPE = 'OFFSET_VARIANCE'
FRMTIME  = value of the frame integration time
```

Additional administrative info related to the EMCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers (e.g. window position and size).

2.7 pn Imaging packets

These packets are characterised by :

an APID of EPCS [Ref. P3 5.8.2],
a Packet Type of 15
a Packet Subtype of 1
a SID of **1** [Ref. P3 5.8.2.11]

There is a separate stream of Imaging packets for *each* quadrant operated in an imaging readout option (either full frame, large window or small window) in Imaging Mode [Ref. P1 section 3.4.2].
Note that in the case of the small window option, the chip in which the small window is located will be read more frequently than the other two in the same quadrant, but its data will
be interleaved in the same packet [Ref. P3 section 5.5.2.1]

2.7.1 Telemetry format

For a complete reference see [Ref. P3 5.8.2.11/12].
For what concerns here it has to be noted that the original stream of photons falling on the given CCD with the following characteristics :

An event e_i at time t_i with energy E_i at CCD pixel position x_i,y_i

is detected during a given frame of a given cycle (the frame is relevant to the integration period of the given CCD, while the cycle is relevant to the 4 quadrants [Ref. P1 3.4.2.7/10], with the exception of the small window option when the cycle is now relevant to 1 quadrant [Ref. P1 3.4.2.13 as amended by item 15 in Addendum]). In what follows events (and associated data formats) in a different cycle/frame will be identified with a different lowercase letter, e.g. a_{Ai},$b_{Ai}c_{Ai}$ ... with the index i running continuously from 1 onwards. The same letter a,b,c... is used to index items related to the specific cycle (where f might be used to index a generic cycle). The index A (upper case letter, one of A,B,C) is used to differentiate data coming from one of the three chips in a given quadrant.

An event e_i is in general produced by *one* X-ray *photon* , but might also correspond to a *pixel hit by a cosmic ray*, or to a *pile-up of more events* . Moreover, one X-ray photon can be **split** across more than one pixel and give rise to more than one event. No on-board reconstruction takes place, therefore such events must be recombined on the ground.

The data stream is a sequence of events e_i (in the sequence $e_{Ai}..e_{Ai+n}$ $e_{Bj}..e_{Bj+m}$ $e_{Ck}..e_{Ck+p}$ i.e. with up to three chunks of events coming from each of the three chips A,B,C in a quadrant; the number of events for the three chips n,m,p are not predictable a priori) interleaved with one time info per cycle T_f (preceding the events of the relevant cycle) and one count info C_{f*} (every
n cycles, with n programmable). C_{f*} will be located after its time info T_{f*}.
In the case of the Small Window option the CCD with the window is read out faster which might results in more events from that CCD in a cycle. Also the data from that CCD may appear in two separate chunks of events interleaved with the other chips [still true ? Ref. P3 pag. 28] e.g. if chip C contains the window the sequence is $e_{Ci}...e_{Ai}...e_{Ci}...e_{Bi}....$

Data follow without solution of continuity from one packet to the next.
The resulting data stream will be (full frame or large window case shown):

Packet 1: H_a a_{A1} ...a_{B1}...a_{C1}...a_{Cp} T_a b_{An+1} ...$b_{Cq}T_b$... .T_c C_{t*} ..
Packet 2: H_e e_{w+1} ...e_x T_e f_z... ... ...

Each packet is preceded by an header (called ITB-header) H_t which identifies the quadrant and readout mode.

About the content of the 32-bit event format e_i [Ref. P3 quoted] one shall note that :

- There is no event time t_i in it. The only timing information is the cycle time t_f contained in the time info T_f.
- the coordinates x_i,y_i (referred to as *line* and *column identifier*) are *chip detector coordinates*
- the energy information Qi is relevant to the event (not to the photon in the case a photon is split among more events)
- each event is tagged with the identifier of the CCD in which it was generated

About the content of the header, time info and count info one shall note that they contain information which either :

- is used to identify the packet (for data sorting) (the quadrant id ; however if sorting of the event on a *one file per CCD* basis is desired the CCD id in the event format shall also be used).
- is used to time tag the cycle and varies monotonically from one cycle to the next (the cycle time T_f in the time info is the end of the cycle [Ref. P3 5.5.3.1]
- are of auxiliary nature and may vary randomly from one frame to the next (the various *counters* and the *common mode mean* in the count info)
- are of auxiliary nature and should not vary from one frame to the next within the same exposure (the *CCD mode* fields in the ITB-header)

2.7.2 Reprocessing

The primary purpose of the reprocessing is to regenerate the original event stream in the readily usable form of an event list, with its scientifically usable information i.e. the best estimate of time t_i , of energy E_i and the CCD pixel position ***with a suitable orientation (i.e. unique for all chips in the focal plane).***
This purpose is fulfilled by the generation of a reformatted event list file (ELF).
However, since this might involve reconstruction of split events, which is not an easily defined algorithm, it is suggested to divide the reprocessing in two stages : the first stage preserves all events and creates a raw event list file (RELF), while the second stage (which could be deferred to the scientific analysis stage) recombines split events (from the RELF it generates the final ELF).

As a secondary purpose it should be considered the production of an ancillary science file (ASF) containing the frame-related information which does not fit the reformatted event list.

It is suggested to create a set of separate files (event list file and ancillary science file) per quadrant. The ASF has only sense if it is created on a quadrant basis. Separate CCD event list files might make sense only in the case of the Small Window option but are not recommended.

The following is the suggested reprocessing :

- as a prerequisite only the relevant packets shall be processed, extracting them from the overall stream using APID, packet type, subtype and SID.
- the *quadrant id* in the *ITB header* in each packet shall be tested, and the events contained in the packet shall be attributed to the relevant event list file according to the quadrant id.
  ~~***Alternately in the small window case*** also the *CCD id* of each event shall be tested, and the event attributed to a separate file according to the quadrant/CCD combination.~~ The CCD or quadrant id shall be saved in the event file list administrative info.

- the commandable parameters (*CCD mode*) in the ITB header of the first packet of a given CCD stream shall be saved in the event list file administrative info.
  ~~***In the case*** of a separate file according to the quadrant/CCD combination, only the CCD mode relevant to one CCD shall be saved to the file pertaining to such CCD.~~
- the value of such parameters in subsequent packet shall be checked not to change. In the case of a change the current event list file shall be closed, and a new one opened (this is assumed to happen only at exposure boundaries).
- the chip coordinates x_i,y_i of each event shall be converted ~~***preferably***~~ to camera-oriented quadrant detector coordinates X_i,Y_i ~~*or at least* to camera-oriented chip coordinates x_i,y_i (***the choice is essentially driven by the fact there is one event list file per chip or per quadrant***).~~
  (The conversion to camera detector coordinates, or to celestial coordinates should be considered in a later processing and not at this stage)
- each event shall be assigned an event time equal to its cycle time ($t_i =_{df} T_f$), see 1.3.3.
- each event shall be assigned the charge Q_i in the raw event list file.
- the counters and parameters in the count info are saved to the ancillary science file, attributing to them also the time of the cycle associated to the count info.
  It is not clear how to do this association. The documentation is ambiguous insofar time infos are produced at EPEA level with a regular cadence (by cycle), but time infos relevant to empty cycles are discarded at EPDH level. Count info appears each n cycles, and (***TBC depending on software development***) its time info will be included in the telemetry even if the cycle is empty i.e. contains zero events.

The above covers the generation of the RELF. The final ELF might be generated from the RELF applying a ***TBD algorithm*** which identifies events e_i coming from the same photons, and writes to the final ELF a single event e_j with Q_j equal to the sum of the Q_i, $t_j=t_i$, and coordinates derived assuming x_j,y_j corresponding to the best estimates (usually the x_i,y_i of the event with highest Q_i). ***A preliminary algorithm emulating for the pn data the EMCR algorithm is available*** in the program `ground_pi` of the EPOS suite [Ref. 2].
The generation of the RELF shall also reject residual cosmic rays present at quadrant boundaries.

2.7.3 Output file format

2.7.3.1 Event list file

A tabular file with one row per event (or reconstructed event). The following is the event format according to the conventions in sections 1.3.4 and 4.

Same format for RELF and ELF except that column 5 is not present in the RELF.

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	cycle time t_i
2	X	PIXEL	1I	CCD coordinate x_i or X_i
3	Y	PIXEL	1I	CCD coordinate y_i or Y_i
4	ENERGY	CHANNEL	1I	raw or reconstructed Q_i
5	PATTERN	--	1I	flag generated by reconstruction algorithm

At least the following administrative info shall also be present.

SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN = 'EPCS'
QUADRANT = quadrant id (0-3) if one file per quadrant
DATATYPE = 'IMAGING'
CCD0MODE = code for CCD mode of first chip in quadrant
CCD1MODE = code for CCD mode of second chip in quadrant
CCD2MODE = code for CCD mode of thirdchip in quadrant
(code for all three CCDs is equal except in the case of small window)

Additional administrative info related to the EPCS configuration may be present, but its source shall be either the HK data stream or the command history file ***since*** the data are not present in the frame headers (e.g. window position and size).

2.7.3.2 Ancillary science file

A tabular file with one row per set of n cycles (i.e. per count info) with the following format according to the conventions in in sections 1.3.4 and 4

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	cycle time T_i related to C_i
2	NABOVE	pixel	1J	pixels above threshold
3	NDEFA	pixel	1I	pixels read by DEFA
4	NEPDH	pixel	1I	pixels sent to EPDH
5	COM_MODE	channel	1I	mean common mode
6	NDISCLIN	lines	1I	discarded line counter

The administrative information shall be the same as described in 2.7.3.1, assuming a one file per quadrant basis.

2.8 pn Timing packets

These packets are characterised by :

an APID of EPCS [Ref. P3 5.8.2],
a Packet Type of 15
a Packet Subtype of **2**
a SID of **2** [Ref. P1 P3 5.8.2.11]

There is a separate stream of Timing packets for *each* quadrant operated in timing readout option in Timing Mode [Ref. P1 section 3.4.3].
Note that only one chip in each quadrant is read out ; in normal operations all 4 quadrants will be operated, but in case of failures some quadrants might be disabled.

2.8.1 Telemetry format

For a complete reference see [Ref. P3 5.8.2.11/12].
Most considerations about the way the original stream of photons falling on the given CCD is inserted in the telemetry stream are the same as in section 2.7.1, with the following exceptions:

- a cycle is defined for convenience by the readout of 20 integrated strips each one 10-pixel high (that is, one readout of the entire chip or "frame"; note that formerly this mode used 10 strips each 20-pixel high, the change is justified by the fact the source PSF, although greater than 10 pixels, is now located half-and-half across a quadrant boundary) [Ref. P2 7.2.1.5 as amended by Addendum item 28]
- A macroframe is defined as 10 cycles, and there is a time info every 10 cycles
- the count info C_{f*} is sent every n cycles [Ref. P3 pag. 31]
- only one CCD per quadrant is read out, therefore the data stream will look like (if CCD A is read out) :

Packet 1: $H_a\ a_{A1}\ ....a_{An}\ T_a\ b_{An+1}\ ...b_{Aq}T_b...\ .T_c\ C_{t*}$ ..
Packet 2: $H_e\ e_{w+1}\ ...e_x\ T_e\ f_z...$ ... ...

The layout of the 32-bit event format e_i is the same described in 2.7.1 above, however the interpretation of some fields is different:

- Accurate timing information can be derived combining the cycle time t_f contained in the time info T_f with the y_i event coordinate.
- The coordinate y_i is not a spatial coordinate, but it is a counter increasing at each integration of a 10-pixel high window in the vertical direction, corresponding to a time bin, and ranges 0 to 199 in a cycle

About the content of the header, time info and count info the considerations in 2.7.1 above hold (the CCD Mode codes may be used to tell which is the active CCD in the quadrant).

2.8.2 Reprocessing

The same general considerations reported in 2.7.2 above hold, however one shall note that :

- in this mode only one CCD per quadrant is active. Therefore the issue "one file per CCD or one file per quadrant" no longer stands (there is one file), but is replaced by the fact that the administrative info of the file shall note which CCD is active.
- the (only spatially-significant) chip coordinate x_i of each event might however be converted ~~***preferably***~~ to camera-oriented quadrant detector coordinates X_i ~~r *at least* to camera-oriented chip coordinates x_i (this choice remains open, and should be solved ***contextually*** with the one in 2.7.2).~~
- the other coordinate y_i is used together with the cycle time t_f to generate an accurate event time $t_i=T_{start}+T_W\, y_i$ where

  T_{start} is the start of the cycle (while t_f in the time info is the end) where y_i is assumed to be zero

  T_W is the time necessary to expose a window and stack it into one row (it depends on the sizes of the window and is assumed to be about 39 µs).

  the addition shall of course take place with all times in the same time units
- nevertheless the raw values of the cycle time T_f and of the y_i coordinate shall be saved to an event list file too (see below for details).
- the counters and parameters in the count info are saved to the ancillary science file, attributing to them also the time of the cycle associated to the count info.

  Here too is not clear how to do this association. Count info appears each n cycles, and (***TBC depending on software development***) its time info will be included in the telemetry even if the cycle is empty i.e. contains zero events.

Moreover the algorithm to generate the ELF from the RELF is different from the one mentioned in 2.7.2. ***A preliminary algorithm emulating for the pn data the EMCR algorithm is available*** in the program `ground_pt` of the EPOS suite [Ref. 2].

2.8.3 Output file format

2.8.3.1 Event list file

A tabular file with one row per event (or reconstructed event). The following is the event format according to the conventions in sections 1.3.4 and 4. The order and names of the columns should be consistent with the cases of the pn imaging mode and of the MOS timing mode (which has similarities with the pn timing mode). In fact (as in 2.3.3.1) one could even consider to have : a RELF with no event reconstruction and no time reconstruction ; a RELF with time reconstruction but no event reconstruction, a final ELF with full reconstruction.

Column	TTYPE	TUNIT	TFORM	Content
a	`TIME`	`s`	1D	reconstructed time t_i
b	`X`	`PIXEL`	1I	CCD coordinate x_i or X_i
c	`ENERGY`	`CHANNEL`	1I	raw or reconstructed Q_i
d	`PATTERN`	`--`	1I	flag generated by reconstruction algorithm
e	`RAWTIME`	`s`	1D	raw cycle time t_f
f	`RAWY`	`PIXEL`	1I	raw CCD coordinate y_i

The final ELF shall contain (in the order given) at least columns a,b,c,d. The rawest RELF (no reconstruction at all) shall contain columns e,b,f,c. An intermediate RELF with time reconstruction shall contain columns a,b,c. The simplest solution looks like to have only the rawest RELF and the final ELF (the latter will contain reconstructed events, i.e. less events).

Note however it could be advisable to preserve the raw y_i info also in the final ELF (i.e. it should contain columns a,b,c,df), since the most accurate estimate of the event time depends on a fine correction based on y_i-Y_* where Y_* is the best estimate of the X-ray source position (known to the final user, which may want to perform this fine correction).

Concerning administrative info see 2.7.3.1, but change `DATATYPE = 'TIMING'`, specify `CCDMODE` equal to the timing mode code (redundant) and place the `QUADRANT` or `CHIP` keyword according to the fact `X` is in (camera-oriented) quadrant or chip detector coordinates.

2.8.3.2 Ancillary science file

Same layout described in 2.7.3.2, one row per set of n cycles (i.e. per count info).

2.9 pn Burst packets

These packets are characterised by :

an APID of EPCS [Ref. P3 5.8.2],
a Packet Type of 15
a Packet Subtype of **2**
a SID of **3** [Ref. P3 5.8.2.11]

There is a separate stream of Burst packets for *each* quadrant operated in burst readout option in Burst Mode [Ref. P1 section 3.4.4].
Note that only one chip in each quadrant is read out ; in normal operations all 4 quadrants will be operated, but in case of failures some quadrants might be disabled.

2.9.1 Telemetry format

For a complete reference see [Ref. P3 5.8.2.11/12].
Most considerations about the way the original stream of photons falling on the given CCD is inserted in the telemetry stream are the same as in section 2.7.1, with the following exceptions:

- a cycle is defined by the readout of one "burst" of data
- the count info C_{f*} is sent every n cycles [Ref. P3 pag. 32]
- only one CCD per quadrant is read out, therefore the data stream will look like (if CCD A is read out) :

Packet 1: H_a a_{A1} ....a_{An} T_a b_{An+1} ...$b_{Aq}T_b$... .T_c C_{t*} ..
Packet 2: H_e e_{w+1} ...e_x T_e f_z... ... ...

The considerations given in 2.8.1 about the interpretation of the layout of the 32-bit event format e_i and about the content of the header, time info and count info apply here too. The way the y_i coordinate is used is a however different in detail (see 2.9.2 below).

2.9.2 Reprocessing

The same general considerations reported in 2.8.2 above hold with the only difference that the algorithm to reconstruct the time of an event using the y_i coordinate for the Burst mode is :

- $t_i=T_{start}+T_s\,(y_i + 200)$ where
  T_{start} is the start of the cycle (while t_f in the time info is the end)
  T_s is the time to shift one row during the continuous sample phase (assumed to be 1 µs)

In addition the algorithm to generate the ELF from the RELF (split event reconstruction) is the one mentioned ***in 2.7.2 for imaging mode*** (and ***exemplified by*** program `ground_pi` of the EPOS suite [Ref. 2].

2.9.3 Output file format

2.9.3.1 Event list file

The same format described in 2.8.3.1 can be used (presence of y_i coordinate might be desirable), with the exception that in the administrative info one has `DATATYPE = 'BURST'`, and specifies `CCDMODE` equal to the burst mode code (redundant).

2.9.3.2 Ancillary science file

Same as described in 2.8.3.2

2.10 pn Pixel Characteristics packets

These packets (formerly called Offset packets in [Ref. P1 and P2]) are generated at EPDH level assembling information sent from EPEA over the LBR interface, and are characterised by :

an APID of EPCS [Ref. P3 5.8.2],
a Packet Type of 15
a Packet Subtype of **4**
a SID of **6,7 or 8** [Ref. P3 5.8.2.11], respectively for Offset/Bad pixel map, Pixel Noise map, Discarded Line map.

There is a stream of Offset packets for *such* quadrant for which the EPCS is in Pixel Characteristics Determination Mode [Ref. P1 section 3.5.6 as amended in Addendum item 41].
This mode is likely to be used regularly (e.g. as part of the observation setup procedure). Its data will be used by SOC to adjust or verify the offset setup. As such they may not be relevant for Guest Observers (but inclusion in ODF could be considered as an option).
An exception might be represented by the Discarded Line option, if it needs to be done regularly and frequently (TBC within AIT). Baseline is not more frequently than 1 hr if at all during exposure.

2.10.1 Telemetry format

For a complete reference see [Ref. P3 5.8.2.11/12].
In this mode the information transmitted is not directly related to detected photons but is a report of a selected set of values stored in the EPEA for all individual pixels.

This operation is made once for a given activation ("shot") of the Offset Determination Mode, which involves the transmission of one frame of data for the entire set of 4 quadrants.
[Ref. P3 pag. 41] is inconsistent with the rest of the documentation, including [Ref. P3 pag. 30] referring to a single CCD at a time. It will probably be corrected
This mode will always be done in one shot, as it is dumping pre-calculated information.

The datastream for one quadrant are either :

- Offset/Bad Pixel case : merged offset and status values of single pixels, N=3∞200∞64 values $d_1..d_N$; these values are sent in blocks of 59 pixels (except the last which can be shorter) preceded by a T-header, and 4 blocks are assembled in a packet, as specified in [Ref. P3 5.4.5 and 5.8.2.11].
- Pixel Noise case : noise values of single pixels, N=3∞200∞64 values $d_1..d_N$ in the same order used for the Offset/Bad Pixel data. The noise value (variance σ) will be multiplied by 10 to preserve precision.
- Discarded Line case : the discarded line counter for each line, N=3∞200 values $d_1..d_N$ arranged in blocks in the same way used for the Offset/Bad Pixel data, and followed by an additional EPDH time T (32-bit with 1 s resolution) [Ref. P3 5.8.12.1]

There is no time info for the other kind of blocks ???

Therefore there will be a fixed number of packets per shot M, as a function of N.

The N offset values d_i follow in blocks from one packet to the next, and the resulting data stream will be, after the standard SID :

Packet 1: $H_1\ d_1\ ..d_{59}.H_2\ d_{60}.\ .d_{118}.H_3 d_{119}..d_{577} H_4 d_{178}.d_{536}$

Packet 2: H_4 d_{237} ....d_r

...

Packet M: H_M d_{u+1} ... ... d_N (T)

The headers H_j are so-called T-headers, and contain the CCD id and other information useful to verify the integrity of the data stream.

The data objects are either [Ref. P3 5.8.2.12.4/5/6]:

- Offset/Bad Pixel case : d_i include a 4-bit pixel status s_i and a 12-bit offset value o_i [defined in Ref. P3 5.8.2.12.1]
- Pixel Noise case : d_i is a 16-bit pixel noise value scaled by 10
- Discarded Line case : d_i are 16-bit counters

2.10.2 Reprocessing

The logical data structure produced by the Pixel Characteristics Mode is one of the following (in all cases the T-header info is used only to tell to which CCD in the quadrant, and to verify the sequence of data blocks):

- Offset/Bad Pixel case : the ensembe of two maps of pixel status and offset value. Each map covers one quadrant, in the form of one 192∞200 image.
  The reprocessing shall therefore reconstruct the data stream d_1...d_N in the form of two quadrant images $s_{1..N}$ and $o_{1..N}$ (the s_i and o_i values shall be converted into 16-bit values).
- Pixel Noise case : one map of pixel noise value. Each map covers one quadrant, in the form of one 192∞200 image.
  The reprocessing shall therefore reconstruct the data stream d_1...d_N in the form of one quadrant image (the values shall be converted into standard 16-bit values).
- Discarded Line case : the ensemble of three 200-element arrays of discarded line counters.
  The reprocessing shall therefore reconstruct the data stream d_1...d_N in the form of three arrays in a tabular form.
  In addition the reference time T shall be saved in the administrative info of the tabular file.

There is no information in the first header H_0 which is relevant for the administrative info (all other information is used only for checks during the processing).
There is no ancillary science file associated to the header content.

2.10.3 Output file format

2.10.3.1 Status and offset image files

Separate files shall be created with the pixel status and offset values.

These two files are frame image files. Each one is a 2-d image with 192∞200 pixels of 16-bit width. (In the case not considered here one wishes separate images per CCD, one might consider either separate files per CCD, or preferably stacking three 64∞200 images into a 3-d image, or concatenating the three images as image extensions.).

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EPCS'
QUADRANT =  quadrant id (0-3) if one file per quadrant
CHIP     =  CCD id (1-12) only if one file or extension per CCD
DATATYPE = is unnecessary, implicit being this an image file
BUNIT    = 'PIXEL STATUS' or 'OFFSET VALUE'
REFTIME  =  value of the time in the first header
```

Additional administrative info related to the EPCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers.

2.10.3.2 Noise image files

This is a frame image file with a 2-d image with 192∞200 pixels of 16-bit width. (see 2.10.3.1 for other considerations).
All considerations above in 2.10.3.1 apply with the only difference that in the administrative info one has of course BUNIT= 'PIXEL NOISE'

2.10.3.3 Exposure map files

A tabular file with one row per shot (usually there will be a single shot). The following are the shot formats according to the conventions in sections 1.3.4 and 4.

Column	TTYPE	TUNIT	TFORM	Content
1	TIME	s	1D	time
2	CCD*a*MAP	LINES	200I	CCD *a* counters $d_{1..200}$
3	CCD*b*MAP	LINES	200I	CCD *b* counters $d_{201..400}$
4	CCD*c*MAP	LINES	200I	CCD *c* counters $d_{401..600}$

a,*b*,*c* are the relevant CCD number (1-12).
In the case the file consists ***always*** of a single shot, one might dispense with the TIME column and place the value in the administrative info.

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EPCS'
QUADRANT =  quadrant id (0-3) if one file per quadrant
DATATYPE = 'DISCARDED LINE COUNTER'
DURATION =  duration of the interval to which file refers
            (warning ! this information is not present in telemetry now !
```

Additional administrative info related to the EPCS configuration may be present, but its source shall be either the HK data stream or the command history file *since* the data are not present in the frame headers.

2.11 pn Diagnostics packets

These packets are characterised by :

an APID of EPCS [Ref. P3 5.8.2],
a Packet Type of 15
a Packet Subtype of 3
a SID of **5** [Ref. P3 5.8.2.11]

There is a stream of Diagnostics packets for *such* quadrant for which the EPEA is operating the Common Mode Filter in Transparent Mode , ie. the EPCS is in CCD Diagnostics Mode [Ref. P1 section 3.5.7].
This mode will be used for checkout and diagnostics (in flight presumably during set-aside observations requested by SOC) , and its data will not be relevant for Guest Observers.

2.11.1 Telemetry format

For a complete reference see [Ref. P3 5.8.2.11/12].
In this mode the information transmitted is not directly related to detected photons but to the charge content of each pixel (including noise etc.).

The basic operation is made L times for a given activation ("shot") of the EPEA Transparent Channel Mode, which involves the transmission of L pseudoframes of data for one selected CCD in each quadrant, being read-out with any of the imaging, timing or burst options. L may range 1 to 100.
A pseudoframe is defined as a *group of four rows* from the given CCD. Scanning of an entire CCD is achieved since an exposure will consist of K=50 consecutive shots, advancing to the next 4 rows.
The datastream for one CCD are repeated values of the pixel charge contents, N=KxL∞4x64 values $p_1..p_N$, with a n EPEA header and time info every shot (of L∞4x64 values)
Therefore there will be a fixed number of packets per shot M, as a function of L, and this sequence will be repeated K times.

The L∞4x64 pixel values p_i follow without solution of continuity from one packet to the next, and the resulting data stream (repeated K times, k=1,K) will be :

Packet 1: $H_1\ K_k\ p_1\ p_2 \ldots\ldots p_q$
Packet 2: $H_2\ p_{q+1} \ldots\ldots p_r$
...
Packet M: $H_M\ p_{u+1} \ldots \ldots p_N\ \ T_k$

The headers H_j are D-headers [Ref. P3 5.8.2.12.3] which identify the quadrant and CCD. The K_k is the EPEA header which tells the start and end row of the block. T_k is a standard (EPEA) time info.

2.11.2 Reprocessing

The logical data structure produced by the CCD Diagnostics Mode is a nested sequence of K groups of L image chunks $d_{i..K}$ with dimension 4 rows x 64 columns. From these one may construct a pseudo-sequence of L images (64x200, where 200=Kx4, i.e. a data cube Lx64x200) for the entire CCD (note however that each image will be made by chunks not taken at the same time, but separate by an entire shot). Alternately one may construct K cubes each one Lx64x4, or other arrangements ***according to analysis needs***.

The reprocessing might therefore reconstruct the data stream $p_1...p_N$ in the form of image cubes (each "pixel data format" as described in [Ref. P3 5.8.2.12.3] shall be converted into a 16-bit charge value zeroing the first 4 bits).
Eventual reprocessing to reorder pixels according to detector-oriented coordinates is to be evaluated according to analysis needs.

The extraction of header/trailer information is concerned only with decoding the quadrant and CCD info to tell which CCD is being diagnosed, and to extract the time info. The other parameters including the EPEA header are used only to check the correct reconstruction of the data stream. There is no ancillary science file associated to the header content.

2.11.3 Output file format

2.11.3.1 Diagnostic image files

Although not properly "frames" these data can be handled as frame image files. There are several possibilities, one is to use n-dimensional images as data cubes (e.g. `NAXIS=3`, `NAXIS1/2/3 = L,64,200` or permutation thereof; or `NAXIS=4`, `NAXIS1/2/3/4 = K,L,64,4`), another one is to use FITS "image extensions" following a null primary header (e.g. L 2-dimensional extensions each one 64x200 ; or K 3-dimensional extensions each one Lx64x4 ; or other combinations).

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = 'EPCS'
QUADRANT =  is unnecessary since one CCD at a time
CHIP     =  CCD id (1-12)
DATATYPE = is unnecessary, implicit being this an image file
BUNIT    = 'PIXEL CONTENT'
REFTIME  =  value of the time in the first header
```

3 HK data

The only HK telemetry data likely to be of some (if any) relevance for the scientific user are the Periodic Telemetry Reports (Packet Type =1). These data might need to be processed e.g. for inclusion in the ODF, or for archiving, in the form of an House Keeping File (HKF).
Some other telemetry reports, although not directly going into the ODF, might be used for ODF production (this might include command reports if they are needed to derive the instrument configuration information for a given exposure : such information has to be placed somehow in the administrative information of the ELF and ASF descrived above).

3.1 Telemetry format

As the format of all such report is not known in detail at time of writing, only general considerations will be given.
It is assumed at the moment that MOS and pn will produce a stream of Periodic Telemetry Reports with a fixed frequency of 8 s while pn will produce an additional stream with frequency of 64 s. In all cases packets will include a number of parameters with commutation= 1.

3.2 Reprocessing

If the above assumption is correct, the processing to be done shall involve the extraction of an "interesting" subset of HK parameters (this subset should be programmable, e.g. in a table-driven manner). For such parameters conversion into engineering units shall be performed, and the converted values shall be written to a tabular HKF.

3.3 Proposed output format

Tabular files with one row per HK sample (8 or 64 s). There will be as many columns as "interesting" parameters. Two separate files are required, one for 8s and one for 64s Periodic Reports.
In the eventuality of either commutated parameters (occurring at an interval submultiple of 8s) it is also suggested to create a separate HKF for each set of parameters with a given periodicity (this will in both cases simplify subsequent software).

Column	TTYPE	TUNIT	TFORM	Content
1	`TIME`	`s`	1D	sample time t_i
2	`parameter 1`	`it depends`	1*X*	value of given parameter
...	...	...	...	...

X may be I,J or E according to convenience. The dimensionality is always 1.

At least the following administrative info shall also be present.

```
SATELLIT = 'XMM'
INSTRUME = 'EPIC'
CHAIN    = EPIC chain identifier
```

It *might* not be necessary to add administrative info about the instrument configuration in the exposure to which the HKF refers, if such info is already present in parameters included in the HKF (however *if* such is the case, such information will *by definition* remain constant in all data records - which might be a reason to include it *once only* as administrative info).

4 General guidelines

In this sections are listed some of the common assumptions made in writing the previous sections, and some considerations about the possible implementation (e.g. concerning disk file formats).

4.1 Assumptions

It is assumed to have a family of separate files per exposure. Such files shall be formed by a data area and an administrative info section. The structure of the data area shall be as simple as possible (i.e. a tabular format for event lists and housekeeping parameter lists, or a simple image format).

It is assumed that the scientific user needs to have event data in a coordinate system which is not prone to confusion when used, therefore ideally with a clearly defined orientation.

It is assumed that the scientific user needs to have other event data information (e.g. time, energy) in a form immediately usable; also the user needs to have reconstruction of split events.

On the other hand for safety it is recommended that, whenever the original information in the telemetry is modified by processing (i.e. processing other than reshuffling or sorting of data from the "floating" format in the telemetry to an ordered tabular event list, or other than bit masking to get rid of bits used only as markers in the telemetry stream, or to present quantities only in multiple of 1 (or 2) bytes), the user is given also the unprocessed information, usually in the form of two files (a raw and a processed one).
This allows the user to repeat the processing with alternate algorithms when desired.

As a waiver to the preceding recommendation, the re-orientation of the spatial coordinate systems (three paragraphs above) is considered such a simple and reproducible process that it is not necessary to save the raw information.

4.2 Considerations

The following considerations are mainly aimed to discuss the FITS issue, or in general the way data is physically written to disk.

A preliminary consideration is that a telemetry streams contains (for various reasons of optimization of usage of the available space) fields which can be any number of bits, while computers prefer to use fields which are 8-16-32-64 bit long to match the commonly used data types. It is therefore mandatory, and it has been assumed above, that any field which is shorter than one of the above lengths is zero padded to the closest greater length (thus 1 to 7 bitd become at least one 8-bit byte, 9 to 15 bits become at least one 16-bit word, 17 to 31 bits become a 32-bit longword).

As a further consideration one should strongly prefer the usage of signed quantities (therefore in a 16-bit word there are 15 useful data bits, and one sign bit; as a consequence if the original data uses all of 16 bits to express an unsigned value, it is preferred to expand it to a 32-bit signed longword).
As a corollary, since the interpretation of numeric byte fields is not standardized across different hardware platforms, operating systems and programming languages, it seems wise not to use 8-bit quantities, but go straightly to 16-bit quantities which are used as signed integers. An exception could be flag fields which assume only 0-1 values (or anyhow a small set of values), which could remain byte fields.

Finally, given the annoying complications involved in using in the same program short and long integers, one might consider to use always 32-bit integers even when 16-bit ones would suffice. A trade off between space required and simplicity of use shall be made (however I would recommend that, even if the quantities as 16-bit on disk, they are read in by a dedicated routine, which presents them as 32-bit integers to the rest of the program).

The following quotation is from the NOST periodic posting "FITS basics and information" (on Usenet newsgroup sci.astro.fits). The underlining is from the author of this note.

> ***"1.1 What FITS Is***
> *FITS (Flexible Image Transport System) is a data format designed to provide a means for convenient exchange of astronomical data between installations whose standard internal formats and hardware differ."*

The key word (!) in the above is "transport". When it comes to deciding a format for data exchange (as it might be the case of the ODF), usage of FITS is an obvious choice.
In practice if we talk of reformatting a telemetry stream, the above requirement of using only 8-16-32 bit quantities is mandated by the FITS standard, therefore it does not represent an extra load with respect to doing such a conversion to another format.

FITS has however some drawbacks which are : some unnaturality of layout (which derives from its concept of 2880-byte blocks which are usually not commensurable with the more natural data records), some slight size inefficiency (again deriving from the block size and from the usage of ASCII headers), some computing time inefficiency (on those operating systems which do not use the same internal representation as the big-endian IEEE one used by FITS), and the fact that FITS files are not easily extended without rewriting (deriving from the fact the header is located before the data section).

The latter two drawbacks are of no relevance talking of data exchange, and the former two are nowadays of little relevance, when compared with the available world of FITS readers.

The matter is different if we are considering a work format, like the disk file format used by a given analysis package (e.g. for calibrations) or if we consider the archiving of a massive set of data for "private" use (again the case of calibrations).

In the case of an analysis packages both the latter drawbacks are relevant, particularly the one on the non-extendability of the header (e.g. adding extensive `HISTORY` when manipulating a file). In the case of archiving, or on massive usage for a "private" use, also the overhead due to conversion from the FITS internal format to the native binary internal representation of the machine used may become significant.

In some cases a trade-off in this respect is possible if the choice of the operating system is defined a priori. Thus if one is using exclusively Sun or HP platforms, usage of FITS is favoured because the big-endian IEEE format is native (no conversion needed), on the contrary on little-endian platforms (DEC Ultrix, DEC OSF/1, DEC AXP VMS, MS-DOS) byte swaps are needed systematically, and the situation is even worse on some proprietary systems (DEC VAX VMS, and perhaps AXP VMS, IBM VM) when floating point data are considered.

The fact that sections 2 and 3 use a notation similar to the FITS one (e.g. the administrative info parameters are indicated as FITS header keywords, or the columns of a table as columns in a FITS `BINTABLE`) does not imply an apriori choice for FITS.
Definition of different more effective formats is possible (as most major analysis packages do). In fact my EPIC simulation EPOS software [Ref. 2] uses a peculiar "XAS" format [Ref. 2 and 3]. In such cases one should however consider as a requirement to be one-to-one mappable to FITS for data exchange purposes.

Even in the case one decides to use FITS, one should consider a set of guidelines to prevent to be overwhelmed by the plethora of *possible* formats *allowed* by the very latitudinary and flexible FITS standard (particularly in the area of binary tables) and to avoid to complicate one's software life. The basic rule should be the KISS rule ('*Keep It Simple Stupid*'), some of whose corollaries are :

- one should use only the following HDU types : basic image HDU, possibly image extensions, *single* binary table extension.
- Image data (in the present context limited only to frame images in diagnostic mode) should be saved to plain FITS file consisting of a *single* basic HDU (primary array). In the case one desires to put more *associated* images in the same file (as the case of multi-shot diagnostic mode exposures discussed in section 2.5) one has two choices : make a single image with three dimensions, stacking the usual 2-d images with additional dimension time (or shot number); or put the additional images in *image extensio*n HDUs, which are reasonably simple.
- in no other case files shall contain more than one complete HDU, i.e. one either has a valid primary array (basic image HDU) and no extensions, or one has a primary HDU without a data array, followed by a *single* `BINTABLE` extension.
- such tables shall usually be very simple, i.e. consisting of n rows with m columns of *depth one*. Format 1J (32-bit integers) or 1E (32-bit floating point, for computation results) are preferred over 1I or 1B (16- and 8-bit values) for the reasons given above.
- in particular cases one may consider a column of depth either than one (if one wants to store an array of *known constant width* like the MOS energy information). The "variable array" form shall be *avoided*.

- in case of binary tables, the primary HDU shall contain only the mandatory keywords, all other header keywords shall go only in the (only) extension header.
- the number and format of header keywords shall be kept as simple as possible, in order to avoid administration overheads. The information described in sections 2 and 3 as "administrative info" shall map to header keywords.

Last but not least, one shall not forget Ockham's razor : "File structures *non sunt multiplicanda praeter necessitatem*".

5 References

General references

[1] NASA NOST, Implementation of the Flexible Image Transport Systems (FITS), latest issue as available on line at `href=http://www.gsfc.nasa.gov/astro/fits/fits_home.html`

[2] L.Chiappetti, EPOS : EPIC Observation Simulations.
Issue 2.1 - 28 Aug 95 - EPIC-IFC-TN-006

[3] L.Chiappetti, XAS software information is available at :
`http://sax.ifctr.mi.cnr.it/Xashelp` (user manual)
`http://sax.ifctr.mi.cnr.it/Sax/xasblurb.html` (basic concepts)
Further extensive documentation is available on request.

[4] e.g. ESA XMM science operations centre implementation plan (SIP) and XMM science operations centre implementation requirements (SIRD)

EPIC applicable documents

[M1] LABEN, Requirement Specification for the EPIC MOS camera system (EMCS), EPIC-LAB-SR-002, Issue 3, February 95
and related addendum by EST, Jun 95

[M2] LABEN, Requirement Specification for the EPIC MOS data handling (EMDH) unit, EPIC-LAB-SP-002, Issue 3, March 95
and related addendum by EST, Jun 95

[M3] SAP Saclay, Unit Requirement Specification for the EPIC MOS Control and Recognition Unit, EPIC-SAP-RS-001, Issue 3, Jan 95

[M4] Leicester University, Design Document of the EPIC MOS Analogue Electronics (EMAE), EPIC-LUX-DD-YYY, Issue 1, May 95

[M5] EST, EMCS Electrical I/f Specification, EPIC-EST-SP-001, Issue 3, Aug 96

[P1] LABEN, Requirement Specification for the EPIC pn camera system (EPCS), EPIC-LAB-SR-003, Issue 3, April 95
and related addendum by EST, Jun 95

[P2] LABEN, Requirement Specification for the EPIC pn data handling (EPDH) unit, EPIC-LAB-SP-004, Issue 3, April 95
and related addendum by EST, Jun 95

[P3] EST, EPCS Electrical I/f Specification, EPIC-EST-SP-002, Issue 2, Sep 96

[R1] M.Boer, Specifications for the Archiving of the EPIC Radiation Monitor Calibration Science Telemetry. EPIC-CES-SR-002 Issue 1 - 19 Dec 95