Guideline to the PETLINK™ Proposal

21-Jun-2018

Revision J2

PETLINK™ – A proposed digital interconnect standard for data acquisition in nuclear medicine.

A special thanks to the many kind people who have contributed to this document over an estimated 20 year period – 1998 to 2018. These include: L. Byars, J. Reed, P. Luk, E. Breeding, J. Bao, J. Everman, A. Moor, I. Huff, M. Casey, R. Nutt, & B. Bendriem. WFJ

William F. Jones
Senior Key Expert
Siemens Medical Solutions USA, Inc.
HC DI MI PLM-R&D RCC
810 Innovation Drive
Knoxville TN 37932 USA
voice: 865 218 2254
fax: 865 218 3010
email: william.f.jones@siemens-healthineers.com
web: www.usa.siemens.com/healthineers.com

Table of Contents:

Revision History:

Rev:	**Date;** Brief Summary of Changes;
G9	**30-Jan-2006;**
H1	**23-May-2007;** TAG 3 Formats Expanded; TAG 4 Acquisition Flag; G-LINK History;
H2	**1-Nov-2008;** IDLE; 30-bit Bin-Address Field; TAG 1 Dead-Time; Tag 2 Horizontal Bed Moving Bit; TAG 3 Exp. Format #2;
H3	**29-Nov-2010;** Acquisition Flag Packets: Time Sync. & Redundant Insertion;
H4	**23-Feb-2012;** Add "CONFIDENTIAL" to Each Page;
J1	**2-Jan-2013;** Remove "CONFIDENTIAL"; Add Filler Packet; Add 2 TOF bits; Add Tag_56PL;
J2	**21-Jun-2018;** Add Ethernet; Reformat 64-bit detector- pair packet for longer PET Axis (AY/BY), more XE bits, more TOF (TF) bits, & more Energy (AE/BE) bits; The infrequently utilized DOI fields (AI/BI) are to be shared; Report 10GbE impact on Lost Event Counter Tag Packets; Remove Flow Control / SUSPEND discussion.

This document is an evolving guideline for a proposed digital interconnect standard for data acquisition in nuclear medicine - PETLINK™. This proposal is intended for PET and the PET portion of PET/CT, PET/MR, PET/SPECT, etc. Primarily PETLINK describes digital formats for real-time data acquisition, processing and storage of coincidence-event data. A brief description of the interfaces used is also provided.

Note: The use of the information in this document is for research purposes only and is not for clinical use.

1. Hardware Interfaces

In part, our PETLINK-compatible data-acquisition hardware is required to stream coincidence-event data in real time from our custom-designed-for-PET data-processing hardware (e.g. GIM, etc.) to our server-class data-acquisition PC (e.g. ACS, MARS, etc). For PETLINK, we prefer using standard, high-speed interfaces for that streaming connection. Older design efforts focused primarily on Fibre Channel (FC). [See Fibre Channel web site: http://fibrechannel.org/]. Starting in 2018, our PETLINK-compatible designs focus on Ethernet – e.g. 10GbE. [See Ethernet description: https://en.wikipedia.org/wiki/10_Gigabit_Ethernet].

1.1. Older Stream Protocol for Fibre Channel

The primary interface focus for PETLINK from roughly 1999 through 2017 has been Fibre Channel. Unlike more complex FC protocols, the PETLINK FC stream protocol emphasizes simplicity. For us this has meant using 10-bit transmission characters generated by FPGA-resident SER/DES – e.g. Xilinx Rocket I/O – and characters sent via a simplex fiber-optic communication link – e.g. Avago AFBR-57R5AEZ. No FC frame delimiters are used. The primary FC ordered set is the IDLE primitive. IDLE is used to maintain interface sync during periods of non-peak throughput. For 64-bit packet transmission, contiguous transmission characters forming 64-bit data packets are inserted into this IDLE stream. To help ensure receiver synchronization, the PETLINK transmitter should ensure that IDLE transmission characters always make up some fraction (however small) of the stream - e.g. no fewer than 1 IDLE ordered set for every 128 set-of-4 transmission characters for data even during periods of full-rate data transmission.

1.2. Newer Stream Protocol for Ethernet

Starting in 2018, newer PETLINK data-acquisition systems from Siemens use 10GbE instead of FC. At least for the short term, this change eliminates the need for any custom-designed-for-PET data acquisition hardware located within the data-acquisition PC – eliminating also any custom software drivers for the PC OS. Our 10GbE implementation for PETLINK makes use of the Ethernet standards using TCP/IP sockets and full-duplex fiber-optic cabling. Our plans are to use 10GBASE-SR transceivers – e.g. Finisar FTLX8574D3BCV – and OM3 cabling. We also plan to support IP “Jumbo” frames.

2. 64-bit Packet Format

2.1 Following is a 64-bit packet format for detector pair and tag packets:

BIT#	First 32-bit word		BIT#	Second 32-bit word	
0-5	AX0-5	(TW0-5)	0-5	BX0-5	(TW16-21)
6	AX6, XE6	(TW6)	6	BX6, XE7	(TW22)
7	AX7, AE7	(TW7)	7	BX7, BE7	(TW23)
8-14	AY0-6	(TW8-14)	8-14	BY0-6	(TW24-30)
15	AY7, AE6	(TW15)	15	BY7, BE6	(TW31)
16-18	XE0-2	(TW32-34)	16-18	XE3-5	(TW44-46)
19-21	AE0-2	(TW35-37)	19-21	BE0-2	(TW47-49)
22-24	AI0-2, AE3-5	(TW38-40)	22-24	BI0-2, BE3-5	(TW50-52)
25-27	TF0-2	(TW41-43)	25-27	TF3-5	(TW53-55)
28	TF6		28	TF7	
29	TF8		29	Reserved	(Typ. Set to 0.)
30	Tag_64		30	Prompt/Tag_56PL	
31	PS0 = 0		31	PS1 = 1	

Where:

AX, BX Transaxial Head Detector Index. [A Head is a "right section" of the full cylindrical detector array.] Originally two 8-bit fields. Upper two bits may now be repurposed to allow optional expansion of XE field and/or AE/BE fields where needed.

AY, BY Axial Head Detector Index. A pair of 8-bit fields. Optionally support two AE/BE bits.

XE Transaxial Encoding. Allows more optimal use of AX/BX bit-fields. Originally a single 6-bit field. Now optionally expanded to a single 8-bit field where needed.

AE, BE Energy Window. Originally two 3-bit fields. Now optionally expanded to two 8-bit fields where needed.

AI, BI Depth of Interaction. Originally two 3-bit fields. Now may be optionally repurposed for use as additional bits for AE/BE where needed.

TF Time of Flight. A 2's complement field with positive nominally towards the "A" crystal. Originally a single 6-bit field, TF0-5. Now optionally, a single 9-bit field, TF0-8, where needed.

TW Tag Word Field: Bits TW0–31 when Tag_64 = 1 & Tag_56PL = 0; Bits TW0–55 when Tag_64 = 1 & Tag_56PL = 1;

Tag_64 Indicates non-event (Tag) 64-bit packet when set to 1; Event 64-bit packet when set to 0. [See also Prompt/Tag_56PL.]

Prompt/Tag_56PL When Tag_64 = 0, Delayed event if set to 0; Prompt event if set to 1.
When Tag_64 = 1, 32-bit Tag packet Payload (TW0-31) if set to 0;
56-bit **Tag** packet **P**ay**L**oad (TW0-55) if set to 1.

PS Packet Sync; PS0 & PS1.

Note that the two Packet Sync bits are for synchronization with the 32-bit word pair. A receiver of this stream should continuously synchronize on PS0 = 0 in the first 32-bit word. Note also the definition of a 64-bit "Filler" Packet – i.e. a packet to be ignored with "all bits set, save PS0" or "FFFFFFFF7FFFFFFF".

A Comment About the 64-bit Filler Packet:

To amplify, this Filler Packet has been defined specifically to aid in the clean termination of packet (data) acquisitions. Typically when a data acquisition is terminated due to elapsed time, such Filler Packets are employed to ensure that any data buffers which may be only partially filled within DMA hardware are properly flushed. This method ensures that whatever (non-filler) packet data had already been received by the elapsed time mark is actually delivered out via DMA into the PC operating system (e.g. Windows) environment - e.g. into a list-mode file. In a typical usage, the user might encounter a few hundred such Filler Packets at the end of a list-mode file. As mentioned, this Filler Packet is intended to be nothing more than a place holder, contains no useful information and may be readily ignored. As of 23-Aug-2012, the intent is for such Filler Packets to only be used with the 64-bit detector-pair packet format described in this Section 2.1.

2.2 Following is a 64-bit packet format for absolute bin address and tag packets:

BIT#	First 32-bit word	BIT#	Second 32-bit word
0-15	BA0-15 (TW0-15)	0-15	BA16-31 (TW16-31)
16-19	BA32-35 (TW32-35)	16-19	BA36-39 (TW44-47)
20-27	SF0-7 (TW36-43)	20-27	SF8-15 (TW48-55)
28-29	Reserved (Typ. Set to 0.)	28-29	Reserved (Typ. Set to 0.)
30	Tag_64	30	Prompt/Tag_56PL
31	PS0 = 0	31	PS1 = 1

Where:

BA	Bin Address Field, Bits 0-39. Projection Space Index.
SF	Scale Factor, Bits 0-15. For, Normalization, Dead-time, etc.
TW	Tag Word Field: Bits TW0–31 when Tag_64 = 1 & Tag_56PL = 0; Bits TW0–55 when Tag_64 = 1 & Tag_56PL = 1;
Tag_64	Indicates non-event (Tag) 64-bit packet when set to 1; event 64-bit packet when set to 0. [See also Prompt/Tag_56PL.]
Prompt/Tag_56PL	When Tag_64 = 0, Delayed event if set to 0; Prompt event if set to 1. When Tag_64 = 1, 32-bit Tag packet Payload (TW0-31) if set to 0; **56**-bit **Tag** packet **P**ay**L**oad (TW0-55) if set to 1.
PS	Packet Sync; PS0 & PS1.

Note that the two Packet Sync bits are for synchronization with the 32-bit word pair. A receiver of this stream should continuously synchronize on PS0 = 0 in the first 32-bit word.

3. 32-Bit Packet Format

3.1 Overview

```
MSB|                                     |LSB
   TXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX      T = Tag Bit
                                                  (1-Tag/0-Event)
                                                X = Not Restricted

   0XXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX      Event Packet
                                                  (Not a Tag Packet)

   10XX XXXX XXXX XXXX XXXX XXXX XXXX XXXX      TAG 1: Time Marker/
                                                   Dead-Time Tracking

   110X XXXX XXXX XXXX XXXX XXXX XXXX XXXX      TAG 2: Gantry Motions
                                                   & Positions

   1110 XXXX XXXX XXXX XXXX XXXX XXXX XXXX      TAG 3: Patient
                                                   Monitoring:
                                                   (Gating/Physiological
                                                   /Head Tracking)

   1111 XXXX XXXX XXXX XXXX XXXX XXXX XXXX      TAG 4: Control /
                                                   Acquisition
                                                   Parameters
```

3.2 32-Bit Packet Format Details

Event Packet: (Most Significant Nibble = 0uuu)

PET Bin-Address Event Packet

```
0PBB BBBB BBBB BBBB BBBB BBBB BBBB BBBB
  |29                                0|
```

P: Prompt (1 - Prompt; 0 - Delay)
B: Bin Address: 0-29.

Note that older PET systems (< 1998?) designated bit 29 (i.e. B29) as a Transmission/Window (W) bit.

SPECT Event Packet (Obsolete)

```
0uuu uuuH XXXX XXXh SYYY YYYY EEEE EEEE
          |6    0|   |6    0| |7      0|
```

u: Undefined (Default: Set to zero.)
H: Head Index (A=0;B=1)
X: Transaxial Xtal Index: 0-6
h: Hit bit (always = 1)
S: Shape bit (Decay: Fast =1 / Slow = 0)
Y: Axial Xtal Index: 0-6
E: Energy Index: 0-7

<u>Position Profile Data (Maintenance)</u>

```
0uuu uHHH BBBB BBBh XXXX XXXX YYYY YYYY
      210 |6    0|  |7     0| |7     0|
```

u: Undefined (Default: Set to zero.)
H: Head number: 0-2 [See 3 H-field bits labelled by "210" above.]
B: Block position: 0-6
h: Hit bit (always = 1)
X: X position data: 0-7
Y: Y position data: 0-7

<u>Energy Discrimination Data (Maintenance)</u>

```
0uuu uHHH BBBB BBBh FFFF FFFF SSSS SSSS
      210 |6    0|  |7     0| |7     0|
```

u: Undefined (Default: Set to zero.)
H: Head number: 0-2 [See 3 H-field bits labelled by "210" above.]
B: Block position: 0-6
h: Hit bit (always = 1)
F: First energy ADC sample: 0-7
S: Second energy ADC sample: 0-7

<u>Crystal Energy Data (Maintenance)</u>

```
0uuu uHHH BBBB BBBh EEEE EEEE YYYY XXXX
      210 |6    0|  |7     0| 3210 3210
```

u: Undefined (Default: Set to zero.)
H: Head number: 0-2 [See 3 H-field bits labelled by "210" above.]
B: Block position: 0-6
h: Hit bit (always = 1)
E: Second ADC sample: 0-7
Y: Y position in block: 0-3 [See 4 Y-field bits labelled by "3210" above.]
X: X position in block: 0-3 [See 4 X-field bits labelled by "3210" above.]

<u>Crystal Time Data (Maintenance)</u>

```
0uuu uHHH BBBB BBBh 0TTT TTTT YYYY XXXX
      210 |6    0|   |6    0| 3210 3210
```

u: Undefined (Default: Set to zero.)
H: Head number: 0-2 [See 3 H-field bits labelled by "210" above.]
B: Block position: 0-6
h: Hit bit (always = 1)
T: TDC value: 0-6
Y: Y position in block: 0-3 [See 4 Y-field bits labelled by "3210" above.]
X: X position in block: 0-3 [See 4 X-field bits labelled by "3210" above.]

3.2.1 TAG 1: Elapsed Time Marker/Dead-Time Tracking (Most Significant Nibble = 10uu)

Elapsed Time Marker

```
100M MMMM MMMM MMMM MMMM MMMM MMMM MMMM
   |28                               0|
```

M: Elapsed milliseconds: 0-28

Dead-Time Tracking (Legacy Block Singles)

```
101B BBBB BBBB BSSS SSSS SSSS SSSS SSSS
   |9         0||18                  0|
```

B: Block Number : 0-9
S: Singles per second (per 0.25 sec. May-02; per 0.125 sec. Aug-07): 0-18

Note: Originally this 19-bit S field was defined to represent singles per second. As of May 2002, a change was implemented by Siemens/CTI to down shift by 2 bits the singles/sec value prior to loading into this field – making the S value represent Singles/quarter-second. To get a value in Singles/sec, simply up shift the S value by 2 bits – i.e. multiply by 4. This down shift was done in order to extend the top-end range supported by this 19 bit field. In other words, later versions of the blocks began to effectively detect more than 500k Singles/sec. This 2-bit-down shift allowed support for singles/sec rates for each block as high as 2.1 M/sec and with a negligible loss in precision. As of August 2007 and along similar lines, another change was implemented by Siemens for some later version systems to instead down shift by 3 bits. So for those later version systems to get a Singles/sec value from the S field, multiply S by 8.

Expanded Format: Dead-Time Tracking

```
101T TTDD DDDD DDDD DDDD DDDD DDDD DDDD
   2 10|25                           0|
```

D: Data Field: 0-25

T: Type field: 0-2 [See 3 T-field bits labelled by "2 10" above.]

T field value (0 – 7; one of eight) indicates type of D field content.

Guidelines: T = 0 indicates Block Singles content.

T = 1 – 5 reserved.

T = 6 indicates Lost (Prompt or Delayed) Event Count Packets inserted by the "Second Lossy Node" - e.g. the PDR card or the PSB card. [Type 6 not needed after 2017 with move to 10GbE.]

T = 7 indicates Lost (Prompt or Delayed) Event Count Packets inserted by the "First Lossy Node" - e.g. the Coincidence Processor or GIM.

For T = 0: Block Singles Type. [See 3 T-field bits labelled by "2 10" below.]

Define whole tag packet as:

```
1010 00BB BBBB BSSS SSSS SSSS SSSS SSSS
   2 10|6     0||18                  0|
```

where: S: Singles per second (per 0.25 sec. May-02; per 0.125 sec. Aug-07): 0-18

B: Block Number: 0-6

Note that this Block Number field is reduced to 7 bits compared to the original 10-bit Block Number for the Legacy case above. Since the range of block numbers reported was always small (< 128), this reduction to 7 bits should be backward compatible.

For T = 6 or 7: Lost Event Counter Type. [See 3 T-field bits labelled by "2 10" below.]

Define whole tag packet as:

```
101T TTrr rrrr LLLL LLLL LLLL LLLL LLLL
   2 10        |19                   0|
```

where: T: Type Field: 0-2. Values of 6 or 7 only.

r: reserved.

L: Lost Event Packets: 0-19. The L value represents a count of event packets observed to be lost during the processing of 1M (or precisely 1048575) arriving event packets. In other words, 1M packets come in and only 1M – L packets are transmitted. After each arriving 1M packets are processed, the loss count (L) is transmitted in this tag packet format. If L is zero, this indicates there were no lost event packets for the 1M arriving event packets which were just processed.

Example 32-bit Type-6 (T=6) Lost Event Tag packet with L = 0: **B800 0000** (hex)

Example 32-bit Type-7 (T=7) Lost Event Tag packet with L = 0: **BC00 0000** (hex)

With this Expanded Format, an opportunity exists to also incorporate information related to lost event packets and to lay a foundation for future expansion for on-line reporting of dead-time issues. In regard to Type-7 (T = 7) Lost Event Counter Tally (LECT) packets, during periods in which data packet rates in the stream exceed the capacity of the GIM-to-PC interface (e.g. 10GbE), packets can be discarded (i.e. lost) due to a lack of sufficient communication bandwidth. To correct for these losses, these specific LECT tag packets report the number of event packets which were discarded. [Note that these packets lost due to reasons of limited bandwidth are not to be confused with those packets discarded as being outside the chosen PET field of view (FOV). Those FOV-discarded packets are not to be included in these specific lost-packet calculations.] These lost-event tag packets can be inserted to allow down-stream processing (whether on-line hardware or list-mode-driven batch software) of the data stream to correct for these losses as part of an over-all dead-time correction. Such tag packets may not be lost during high rate periods if priority is given by each Lossy Node to lose no tag packets at the expense of lost event packets – i.e. always discard event packets before discarding tag packets.

3.2.2 TAG 2: Gantry Motions & Positions (Most Significant Nibble = 110u)

Detector Rotation Position (Legacy)

To be extracted and applied by the Rebinner in PET/SPECT PET mode.

```
1100 0000 BAFF FFFF FFFF FFFF PPPP PPPP
            |13              0| |7      0|
```

P: PET Rotational Position: 0-7 [For PET mode need 0 - 255d for 0-360° - wfj]
F: Full Precision Rot. Position: 0-13 [Useful for SPECT]
A: CW Rotation: 0 [1 = clockwise gantry rotation as viewed from bed side]
B: CCW Rotation: 0 [1 = counterclockwise gantry rotation as viewed from bed side]
Note: (A=0 & B=0) indicates non-rotating gantry.

Head I "A" Radial Position (Legacy)

```
1100 0001 uuuu uuuu uuuR RRRR RRRR RRRR
                       |12            0|
```

u: Undefined (Default: Set to zero.)
R: Radial Position: 0-12

Head II "B" Radial Position (Legacy)

```
1100 0010 uuuu uuuu uuuR RRRR RRRR RRRR
                       |12            0|
```

u: Undefined (Default: Set to zero.)
R: Radial Position: 0-12

Vertical Bed Position

```
1100 0011 uuuu uuuu uuVV VVVV VVVV VVVV
                      |13             0|
```

u: Undefined (Default: Set to zero.)
V: Vertical Position: 0-13

Horizontal Bed Position

```
1100 0100 uuuM HHHH HHHH HHHH HHHH HHHH
               |19                    0|
```

u: Undefined (Default: Set to zero.)
H: Horizontal Position: 0-19 [Note: For VG30 on Nov-2010 and later mCT systems, this 20-bit H value is always a 2's complement number. When H is near zero - e.g. H = -1(dec) or FFFFF(hex), the bed (a. k. a. Patient Handling System or PHS) approaches (or is) fully retracted from the FOV. When H is at its largest negative extent, the bed is fully extended into the FOV. The LSB of H represents 0.001cm of horizontal bed movement. In recent years, the actual precision reported by the bed has been 0.001cm – i.e. to the LSB limit of the H field. Some earlier versions of the PHS had precision limited to 0.05cm – i.e. incrementing/decrementing the H field by 50(dec) only.]
M: Moving. Single Bit. Set to 0 if horizontal bed motion is unchanging. Set to 1 if horizontal bed motion is changing. [This bit was requested but implementation may be pending.]

Example 32-bit Horiz. Bed Pos. Tag packet (hex) with H = 0: **`C400 0000`** `(hex)`

Gantry Left/Right Position (Legacy)

```
1100 0101 uuuu uuuu uuuP PPPP PPPP PPPP
                       |12            0|
```

u: Undefined (Default: Set to zero.)
P: Gantry Left/Right Position: 0-12 [-2287 to 508 decimal in 0.1 mm steps]
(Positive to the left as viewed from bed side of gantry)

<u>Source Axial Position & Rotation</u> (Point Source, Rotating Rod, etc.)

```
1100 0110 AAAA AAAA AAAA RRRR RRRR RRRR
          |11          0| |11          0|
```

A: Axial Position: 0-11
R: Rotational Position: 0-11

<u>HRRT Single Photon Source Position</u>

```
1100 0111 uuuu HHHH AAAA AAAA RRRR RRRR
               3210 |7      0| |7      0|
```

u: Undefined (Default: Set to zero.)
H: Head Number: 0-3 [See 4 H-field bits labelled by "`3210`" above.]
A: Axial Position: 0-7
R: Rotational Position (within the Head): 0-7

3.2.3 TAG 3: Patient Monitoring: (Gating/Physiological/Head Tracking) (Most Significant Nibble = 1110)

<u>Basic Gating/Physiological Format</u>

```
1110 0EEE uuuu uuuu uuuu uuuu uuuu uuuu
      210
```

u: Undefined (Default: Set to zero.)
E: Expansion Format Control: 0-2. [See 3 E-field bits labelled by "`210`" above.]

<u>Expanded Format #0 (Legacy)</u>

```
1110 0000 rrrr rrrr rrrr rrrr GGGG GGGG
      210                     |7      0|
```

r: Reserved. (Default: Set to zero.)
G: Gating: 0-7
E: Expansion Format Control: 0-2. (The 3-bit E field is set to 0 – as shown.)

Here is a refinement of the Gating byte usage – i.e. the set-of-8 G bits shown above:

```
CPDD DDDD
  |5    0|
```

where:

D: Physiological Data (e.g. Respiratory Phase) Field: 0-5
D content only has validity if flag bit P=1.
P: Physiological (e.g. Respiratory) Flag Bit – High True.
C: Cardiac "R Wave" Flag Bit – High True. This bit is intended to be the primary trigger control for driving the automatic FPGA-driven Gating Buffer switching in designs such as the Smart DRAM. [Note: Smart DRAM was an alternate hardware architecture for PET data acquisition that was considered (~2003) but one that did not go into production.]

Example Format #0 32-bit Cardiac R-wave Tag packet commonly used on HiRez: **`E000 0080`** `(hex)`

Expanded Format #1 (Gating)

```
1110 0001 rrrr rrrr GGGG GGGG GGGG GGGG
      210          |15              0|
```

r: Reserved. (Default: Set to zero.)
G: Gating: 0-15
E: Expansion Format Control: 0-2. (The 3-bit E field is set to 1 – as shown.)

This is the proposed usage format for Gating Tag Packet usage – i.e. the set-of-16 G bits shown above:

```
CPPP rrDD DDDD DDDD
 210   |9        0|
```

where:

D: Physiological Data Field: 0-9
D field content only has validity if P field content is nonzero.
r: Reserved. (Default: Set to zero.)
P: Physiological Type Field: 0-2 [See 3 P-field bits labelled by "`210`" above.]
P field value (0 – 7; one of eight) indicates type of D field content.
Guidelines: P = 0 indicates D field has no valid content. (Default value for P.)
P = 2 indicates R Wave Triggers from multiple ECG devices identified by D - e.g. MR-PET.
D = 0 indicates "primary" R Wave Trigger - e.g. MR-supplied ECG.
D = 1 indicates "secondary" R Wave Trigger - e.g. MR-supplied Oxygen Probe.
D = 2 indicates "tertiary" R Wave Trigger - e.g. customer-supplied ECG
P = 1 indicates Respiratory Trigger Content in D.
[For mMR: D=0 for Gate On. D=1 for Gate Off.]
P = 5 indicates Respiratory Phase/Waveform Content in D.
P = 6 indicates Cardiac Phase/Waveform Content in D.
C: Cardiac "R Wave" Flag Bit – High True. This bit is intended to be the primary trigger control for driving the automatic FPGA-driven Gating Buffer switching in designs such as the Smart DRAM. The intent is for the C bit to only be set = 1 when both the P field = 0 and the D field = 0. Also, the intent is for this C bit to only be set = 1 for single ECG environments. For multiple ECG environments - e.g. MR-PET, see P=2 above.

Example Format #1 32-bit Cardiac R-wave Tag packet commonly used on mCT: **`E100 8000`**`(hex)`

Expanded Format #2 (Generic Trigger)

```
1110 0010 rrrr rrrr TTTT TTTT TTTT TTTT
      210           |15             0|
```

r: Reserved. (Default: Set to zero.)
T: Generic Trigger Field: 0-15
E: Expansion Format Control: 0-2. (The 3-bit E field is set to 2 – as shown.)

The purpose of this proposed Generic Trigger tag-packet format is to support generic requirements for stream marking. These Generic Trigger packets are intended to be used to mark the stream for generic events or with generic data. For example:

1. the patient pushes a button (once or many times) such as to quickly generate an inserted tag packet with the 16-bit T15:0 field = 0;

and/or

2. the technician pushes a button (once or many times) such as to quickly generate an inserted tag packet with the 16-bit T15:0 field = 1;

etc.

Note that this Generic Trigger tag packet should not be confused with the TAG 4 Acquisition Flag packet.

Expanded Formats #3 - #6 (Reserved)

Expanded Format #7 (Special Research)

```
1110 0111 rrrr rrrr SSSS SSSS SSSS SSSS
      210           |15             0|
```

r: Reserved. (Default: Set to zero.)
S: Special Research Field: 0-15
E: Expansion Format Control: 0-2. (The 3-bit E field is set to 7 – as shown.)

The purpose of this proposed Special Research tag-packet format is to support “special” requirements for research-oriented stream marking. These Special Research packets are intended to be used to mark the stream for special events or with unusual data. For example:

1. patient blood pressure monitoring generates an inserted tag packet with the two-bit S15:14 field set to 1 and the 14-bit S13:0 field set to represent the current BP reading;

and/or

2. patient temperature monitoring generates an inserted tag packet with the two-bit S15:14 field set to 2 and the fourteen-bit S13:0 field set to represent the current temperature reading.

etc.

The intent is to allow individual PET researchers the freedom and opportunity to employ this specific-yet-flexible tag-packet format – i.e. a format for marking the event stream with as-yet-unidentified time-critical information. Note that this Special Research tag packet should not be confused with the TAG 4 Acquisition Flag packet.

Motion Tracking (Six Tools with Six Degrees of Motion Tracking - Northern Digital Passive Polaris)

```
1110 1TTT DDDV VVVV VVVV VVVV VVVV VVVV
      210 210|20                     0|
```

T: Tool_#:0-2 (One of six platforms or tools tracked for position)
[See 3 T-field bits labelled by "210" above.]
D: Degree: 0-2 (Designate one of eight reported position values -

```
D:  0    1    2    3    4    5    6    7
V:  Q0   Qx   Qy   Qz   Tx   Ty   Tz   Erms)
```

[See 3 D-field bits labelled by "210" above.]
V: Value: 0-20 (Position values - two's complement - See Polaris Manuals)

3.2.4 TAG 4: Control & Acquisition Parameters (Most Significant Nibble = 1111)

Control tag packets are intended for system component control of devices processing the list mode stream. Currently used by the ECAT Rebinner (no longer in production) to control or determine local board reset, mode select, and pass all functions. (See ECAT Rebinner User Guide. Most significant byte = 1111 1100.)

Acquisition parameter tag packets are planned. Acquisition parameter tag packets are expected to be located primarily (but not necessarily) at the start of the list mode stream. Example parameters may include date and time of acquisition start. For absolute bin address streams, the acquisition parameters may include projection space details like span, ring difference, sinogram dimensions, sinogram bin sampling size, etc.

TAG 4: Basic Acquisition Flag Format

```
1111 1111 CCCC CCCC IIII IIII IIII IIII
|31                                  0|
```

1111 Bits 31- 28. Most Significant Nibble is Header defined by PETLINK to indicate TAG 4 packet.

1111 Bits 27 - 24. Second Most Significant Nibble is the Type field for TAG 4 packet. Set to F(hex) for "Flag".

C Bits 23 - 16. Check sum. Sum of 3 other bytes in this 32-bit packet. The inserter of the packet is responsible for calculating and including the content of this check sum field. The reader of this packet is responsible for confirming that the check sum value is consistent with the whole 32-bit packet before accepting this packet as valid. Here are four example 32-bit Flag Tag packets in hex with correct check sum:
TAG 4 (F); Type (F); CS (FF) [FF+00+00=FF]; ID (0000); FFFF0000
TAG 4 (F); Type (F); CS (00) [FF+00+01=100 or 00]; ID (0001); FF000001
TAG 4 (F); Type (F); CS (01) [FF+00+02=101 or 01]; ID (0002); FF010002
TAG 4 (F); Type (F); CS (45) [FF+12+34=145 or 45]; ID (1234); FF451234

I Bits 15 – 0. Flag Identification Field.

Note that the usual techniques may be employed to contain this 32-bit (TW) tag field within a 64-bit packet. Here are some rough suggestions and comments about the use of the I field:

Incrementing Acquisition Flag Packets:

In this approach the 16-bit I field is simply incremented with each subsequent (non-redundant) insertion of an Acquisition Flag packet into the PETLINK stream. For example, the start of the acquisition for the first bed in a multiple bed position study (not continuous bed motion) would have inserted an Acquisition Flag packet with the I field set to 0. Just prior to ending this first bed position acquisition and prior moving the bed to the next bed position, an Acquisition Flag packet with the I field set to 1 is inserted. For the second bed position, this is marked at the beginning with a Flag insertion which has the I field set to 2. This packet is inserted just as this second bed position becomes stationary. Respectively, a Flag packet is inserted with the I field set to 3 at the end of the second bed position acquisition – just prior to moving the bed along to the next stationary position. And so on. [Now (Dec-2009) we begin to restrict this I-field definition. For incrementing and other general use, only the 15 LSB (I14-I0) in the 16-bit I field are allowed such free use with I15 remaining at zero. See following section on time synchronization.]

Time Synchronization via Acquisition Flag Packets:

Requirements have surfaced for time synchronization between PET and MR, for example. For this, the MSB in this 16-bit I field (I15 bit) is determined to be a "Modality" bit. When I15 is cleared to zero, the tag packet is considered as having originated from the PET system – which supports legacy usage. When the Modality, I15 bit, is set to one, the tag packet is considered to have originated from non-PET signaling. For a synchronizing tag packet resulting from a signal from the MR system, the 16-bit I field is to be 1000 xxxx xxxx xxxx - with "x" meaning that the bit state is unspecified here and remains available for user applications. In other words, we designate the 4 MSB (I15-12) in the I field – when set to 1000 (binary) – as designating a time synchronizing tag packet relating to the timing system within the MR companion to PET. This 4 MSB of the I field can be further specified for other applications as needed.

Redundant Insertion of Acquisition Flag Packets (For Fibre Channel Interfaces Only):

[Note: with our move towards Ethernet instead of Fibre Channel in 2018, such packet redundancy issues are no longer critical.]

Good planning dictates that these Acquisition Flag packets should provide for redundancy. Because the fiber-optic Fibre-Channel communication is known to have bit error rates on the scale of 10^{-12}, there is a small but finite chance that inserted packets may be garbled or lost. For the case of coincidence event packets, losing 1 packet out of trillions is hardly significant. By contrast, losing one unique Flag packet in a list-mode stream can mean a lost (if rare) imaging opportunity.

We plan to deal with this possibility by optionally inserting 1, 2 or 4 (identical, redundant and contiguous) packets – but only when the packets are of a crucial nature such as may be the case for these Flag packets. [Note that this non-zero bit error also is the reason for the check sum field as defined above.] In addition, there should be planning in place such that the application software (and even the FPGA hardware) can support multiple, redundant packets for the Flag packet – both at the time of insertion and at the time of responding to stream content. Insertion of multiple, redundant, identical Flag packets into the stream by FPGA hardware should be automatic - resulting in the specified quantity of contiguous Flag packets inserted with no other packet types intervening. From the beginning, application software (or FPGA hardware) which is written (instantiated) to read and respond to this Flag packet content should respond only to the first arrival of a valid packet – i.e. a packet which shows a valid check sum. If other redundant (and valid) identical packets are received contiguously and just after this first valid packet, these subsequent packets (along with all invalid packets) should be ignored.

In other words, if we find we have problems with corrupted packets, we can choose to have multiple, identical, redundant copies of flag packets (with check sum for validation) which are inserted as a contiguous group to mark each intended point of interest in the packet stream. With this redundancy we gain robustness to identify these points of interest even if a few packets are lost or corrupted.

Further discussion (Dec-2009) on the need for such packet redundancy leads to the following comment. Robustness can also be achieved in some cases without requiring multiple, identical, contiguous insertion of flag packets. As an alternate, the system which originally inserts the packet (e.g. GIM) can instead be required to record (in a remotely accessible register) the millisecond elapsed time tick present when this flag packet was inserted. The acquisition system (e.g. ACS) can then be required to read (presumably as the acquisition ends) and store this time offset value for general reference. The thinking is that if such a critical flag packet is lost (and rarely so, it is expected) from the list-mode stream for some reason, an estimate as to its location within the list-mode file can be inferred from knowing the approximate location to within 1 millisecond. Certainly this alternate approach implies that very few (e.g. as in only one) such critical flag packets are inserted per acquisition.

4. 64-bit Tag Packet Format with 56-bit Payload

The purpose of this section is to describe a newer, 64-bit tag packet format – i.e. one which enables a 56-bit payload field (TW0–55) within 64-bit tag packets. This 56-bit payload field is an expansion on the original 32-bit payload limit (TW0–31) as had been the practice with earlier systems.

4.1 64-Bit Tag Packet Format with 56-bit Payload Field: Overview.

```
PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP PPPP
|55                                                                0|
```

Where:

`P` Expanded 56-bit Payload Field for Tag Word (TW) Bits: 0 – 55

In practice, this 56-bit field is to be subdivided as follows:

```
TTTT TTTT DDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDD
|55   48| |47                                                      0|
```

Where:

`T` Type: An 8-bit Tag Packet Type Field within Bits 48 – 55.

`D` Data: A 48-bit Tag Packet Data Field within Bits 0 – 47.

4.2 Block Singles 56-bit Payload Field.

As of 14-Feb-2012, the only 64-bit tag packet which has been defined for using the new 56-bit payload field is the block singles tag packet.

```
TTTT TTTT BBBB BBBB BBBB BBBB SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
|7     0| |15              0| |31                                  0|
```

Where:

`T` Type: `T` = 0;

`B` Block Number: 0-15

`S` Singles per second: 0-31

In general and especially with recent moves to support continuous bed motion studies, a more wide spread use of the 64-bit bin-address and tag packet format is expected. In such instances, this new-style 56-bit payload tag packet may be supported without compromise. However, note that in those instances in which each of these new-style 56-bit payload tag packets for block singles must still be truncated into the old-style 32-bit payload tag packet, the bit loss that is necessary may not allow all bit-field information to be preserved. Such truncation may be required, for example, during the rebinning step when the 32-bit bin-address event packet is still generated from the 64-bit detector-pair event packet format. In such an instance of forced truncation for the B field above, the current plan is that the 10-bit LSB is simply extracted from the above 16-bit B field and is used to fill the 10-bit B field in the old-style 32-bit block singles tag packet. In addition for the case of the S field above, the appropriate portion of the 32-bit S field above is extracted and is used to fill the 19-bit S field in the old-style 32-bit block singles tag packet. With the latest Aug-2007 convention for a 3-bit down shift in the S field, this means that this specific 19-bit portion (bits 3 – 21) of the 32-bit S field above is extracted and used to fill the 19-bit S field in the old-style 32-bit block singles tag packet.

5. TAXIchip Adaptations

[Note that TAXIchip is now a legacy interface for PETLINK.]

Not all PET applications required high speed or large event packet bit count - e.g. ECAT EXACT, HR+. Also, rotating PET systems can restrict the event stream bandwidth available when crossing the rotational boundary - e.g. infrared slip ring of the ART, 40 foot coax of the E.CAM pulley assembly as used by PET/SPECT. Through the 1990's the AMD serial communication chip family, TAXIchip, was used at CTI PET Systems, Inc. for event stream communications. These implementations were always with 4 byte (32-bit) event packets which did not support (serial) tag packets. Later for PET/SPECT scanner types, a 5 byte (40 bit) packet is detailed - see 6.3 below.

5.1 EXACT (921) 32-Bit TAXI Event Packet Format

1st Byte		2nd Byte		3rd Byte		4th Byte	
0-5	MP0-5	0	Prompt	0-4	AD0-4	0-4	BD0-4
		1	Delay				
		2	Multiple				
		3	BR0				
		4	AR0				
		5	BD5	5	AP2	5	BP2
6	BR1	6	AD5	6	AP0	6	BP0
7	AR1	7	Scatter	7	AP1	7	BP1

MP refers to the module pair encoding value.
AD, BD refer to "A" and "B" crystal indexes in the transaxial direction.
AP, BP refer to "A" and "B" crystal-ring ("plane") indexes within the block-pair in the axial direction.
AR, BR refer to "A" and "B" block-ring-to-block-ring indexes in the axial direction.

Note: All successive 4 byte packets are separated by at least one TAXI Command Strobe

5.2 EXACT (922, HR, HR+, ART) 32-Bit TAXI Event Packet Format

1st Byte		2nd Byte		3rd Byte		4th Byte	
0-4	AD0-4	0-4	BD0-4	0	Prompt	0-5	MP0-5
				1-3	AR0-2		
				4-6	BR0-2	6	AD5
5-7	AP0-2	5-7	BP0-2	7	Scatter	7	BD5

MP refers to the module pair encoding value.
AD, BD refer to “A” and “B” crystal indexes in the transaxial direction.
AP, BP refer to “A” and “B” crystal-ring (“plane”) indexes within the block-pair in the axial direction.
AR, BR refer to “A” and “B” block-ring-to-block-ring indexes in the axial direction.

Note: All successive 4 byte packets are separated by at least one TAXI Command Strobe

5.3 PET/SPECT (& LSO-ART) 40-Bit TAXI Event Packet Format

1st Byte		2nd Byte		3rd Byte		4th Byte		5th Byte	
0-7	AX0-7 (TW0-7)	0-7	AY0-7 (TW8-15)	0-7	BX0-7 (TW16-23)	0-7	BY0-7 (TW24-31)	0-1	AE0-1
								2-3	BE0-1
								4	AI0
								5	BI0
								6	Prompt
								7	Tag _40

AX, BX refer to the “A” & “B” crystal index in the transaxial direction.
AY, BY refer to the “A” & “B” crystal index in the axial direction.
AE, BE refer to the “A” & “B” detected photon energy levels.
AI, BI refer to the “A” & “B” “depth of interaction” detector index in the radial direction.

Note: Not all successive 5 byte packets need to be separated by a Command Strobe. Here up to 20 packets of 5 bytes may be adjacent in the TAXI stream - i.e. no more than 20 packets may be transmitted without an inserted TAXI Command Strobe.

5.4 Adapting TAXI to FC – i.e. VME to PC

An investment was made to allow an upgrade for older, VMEbus & TAXIchip compatible tomographs. The goal was to enable the use of the newer PC-based FC systems for limited data acquisition when servicing the older tomograph. This 64-bit detector-pair packet format represents one derived to support data incoming from these older TAXI formats into the newer FC-compatible format. [Two examples of custom PC hardware designed to make this TAXI-to-FC adaptation work are the TFA card and the PDT card - that is, only one such card need be added to the PETLINK & FC-enabled PC, not both.]:

BIT#	First 32-bit word		BIT#	Second 32-bit word	
0-7	AD0-7	(TW0-7)	0-7	BD0-7	(TW16-23)
8-10	AP0-2	(TW8-10)	8-10	BP0-2	(TW24-26)
11-15	AR0-4	(TW11-15)	11-15	BR0-4	(TW27-31)
16-18	MP0-2		16-18	MP3-5	
19-21	AE0-2		19-21	BE0-2	
22-24	AI0-2		22-24	BI0-2	
25-29	Reserved (Typ. Set to 0.)		25-29	Reserved (Typ. Set to 0.)	
30	Tag_64		30	Prompt	
31	PS0 = 0		31	PS1 = 1	

Where:

AD, BD	Transaxial Detector Indexes. [Only bits 0-5 are used within each 8-bit field.]
AP, BP	Axial Intra-Block-Pair Crystal Ring Indexes.
AR, BR	Axial Block-Ring Indexes. [Only bits 0-2 are used within each 5-bit field.]
MP	Module Pair Number.
AE, BE	Energy Window
AI, BI	Depth of Interaction
TW	Tag Word Field: Bits TW0 – 31
Tag_64	Indicates non-event (Tag) 64-bit packet when = 1; event packet when = 0.
Prompt	**Prompt** event if = 1. **Delayed** event if = 0.
PS	Packet Sync; PS0 & PS1.

Note that the two Packet Sync bits are for synchronization with the 32-bit word pair. A receiver of this stream should continuously synchronize on PS0 = 0 in the first 32-bit word.

6. G-Link Adaptations

[Note that G-Link is now a legacy interface for PETLINK.]

Discussions on the G-Link chip sets from Hewlett-Packard led to the following preliminary formats.

6.1 PET/SPECT 32-Bit G-Link Event Packet Format for Bin Address & Tag (Never Implemented.)

1st 16 Bit Word	2nd 16 Bit Word		
0-15 BA0-15 (TW0-15)	0-12	BA16-28	(TW16-28)
	13	W (or BA29)	(TW29)
	14	Prompt	(TW30)
	15	Tag_32	(TW31)

Note: Not all successive double-16 bit word packets need to be separated by Fill Frame. Here up to 50 packets of double 16-bit words may be adjacent in the G-Link stream - i.e. no more than 50 packets may be transmitted without an inserted Fill Frame.

6.2 Rotating Tomograph 48-bit G-Link Event Packet Format (Never Implemented?)

1st 16 Bit Word		2nd 16 Bit Word		3rd 16 Bit Word	
0-7	AX0-7 (TW0-7)	0-7	BX0-7 (TW16-23)	0-2	AE0-2
8-15	AY0-7 (TW8-15)	8-15	BY0-7 (TW24-31)	3-5	BE0-2
				6-8	AI0-2
				9-11	BI0-2
				12-13	Reserved
				14	Prompt
				15	Tag_48

Note: Not all successive triple-16 bit word packets need to be separated by Fill Frame. Here up to 33 packets of triple 16-bit words may be adjacent in the G-Link stream - i.e. no more than 33 packets may be transmitted without an inserted Fill Frame.

6.3 PET/SPECT 64-bit G-Link Event & Tag Packet Formats (Never Reached Product Stage)

Detector Pair & Tag

1st 16 Bit Word	2nd 16 Bit Word	3rd 16 Bit Word		4th 16 Bit Word	
0-7 AX0-7 (TW0-7)	0-7 BX0-7 (TW16-23)	0-2	XE0-2	0-2	XE3-5
8-15 AY0-7 (TW8-15)	8-15 BY0-7 (TW24-31)	3-5	AE0-2	3-5	BE0-2
		6-8	AI0-2	6-8	BI0-2
		9-14	TF0-5	9-13	Reserved
				14	Prompt
		15	Reserved	15	Tag_64

Bin Address & Tag (Never Implemented)

1st 16 Bit Word	2nd 16 Bit Word	3rd 16 Bit Word		4th 16 Bit Word	
0-15 BA0-15 (TW0-15)	0-15 BA16-31 (TW16-31)	0-5	BA32-37	0-2	AE0-2
				3-5	BE0-2
		6-15	Reserved	6-11	TF0-5
				12-13	Reserved
				14	Prompt
				15	Tag_64

Note: Not all successive quad-16 bit word packets need to be separated by Fill Frame. Here up to 25 packets of quad 16-bit words may be adjacent in the G-Link stream - i.e. no more than 25 packets may be transmitted without an inserted Fill Frame.

6.4 "P39" and "Phoenix" 48-bit G-Link Detector-Pair & Tag Packet Format

1st 16 Bit Word

0-7	AD0-7	(TW0-7)
8-10	AP0-2	(TW8-10)
11-13	AR0-2	(TW11-13)
14-15	0	(TW14-15)

2nd 16 Bit Word

0-7	BD0-7	(TW16-23)
8-10	BP0-2	(TW24-26)
11-13	BR0-2	(TW27-29)
14-15	0	(TW30-31)

3rd 16 Bit Word

0-5	XE0-5
6	AE0 (Scatter)
7	0
8-12	TOF0-4
13	Reserved
14	Prompt
15	Tag_48

Note: Not all successive triple-16 bit word packets need to be separated by Fill Frame. Here up to 33 packets of triple 16-bit words may be adjacent in the G-Link stream - i.e. no more than 33 packets may be transmitted without an inserted Fill Frame.

6.5 Adapting G-Link to FC

The "Phoenix" ECAT product required a G-Link to FC conversion via the PSA card. This 64-bit detector-pair packet format represents one derived to support data incoming from these G-Link-enabled coincidence processors into the newer FC-compatible format. [Two examples of custom PC hardware designed to make this G-Link -to-FC adaptation work are the PSA card and the PDT card - that is, only one such card need be added to the PETLINK & FC-enabled PC, not both.]:

BIT#	First 32-bit word		BIT#	Second 32-bit word	
0-7	AD0-7	(TW0-7)	0-7	BD0-7	(TW16-23)
8-10	AP0-2	(TW8-10)	8-10	BP0-2	(TW24-26)
11-15	AR0-4	(TW11-15)	11-15	BR0-4	(TW27-31)
16-18	XE0-2		16-18	XE3-5	
19	AE0				
20-29	Reserved (Typ. Set to 0.)		19-29	Reserved (Typ. Set to 0.)	
30	Tag_64		30	Prompt	
31	PS0 = 0		31	PS1 = 1	

Where:

AD, BD	Transaxial Detector Indexes.
AP, BP	Axial Intra-Block-Pair Crystal Ring Indexes.
AR, BR	Axial Block-Ring Indexes. [Only bits 0-2 are used within each 5-bit field.]
XE	Transaxial Encoding (Module Pair) Number.
AE0	Scatter Bit
TW	Tag Word Field: Bits TW0–31
Tag_64	Indicates non-event (Tag) 64-bit packet when = 1; event packet when = 0.
Prompt	**Prompt** event if = 1. **Delayed** event if = 0.
PS	Packet Sync; PS0 & PS1.

Note that the two Packet Sync bits are for synchronization with the 32-bit word pair. A receiver of this stream should continuously synchronize on PS0 = 0 in the first 32-bit word.