

The purpose of this document is to provide an introductory walkthrough of one aspect of the PETLINK™ Guideline. Here we present examples of various and representative bit fields as they might be used – fields which are currently active in support of the Siemens 4-Ring mCT 64-bit detector-pair packet format. The goal is to help the first-time user gain a better, general understanding of the packet format in use.

A companion example 64-bit list-mode file is provided: **ea4r_64dp_walkthrough.l64**

Two companion example C-code files are provided:

ml_ea4r_64dp_walkthrough_1.c

[Generates the example *.l64 file.]

lmsw64dp_verbose_1.c

[Reports contents of *.l64 file verbosely.]

See also the PETLINK Guideline file.

Overview of list-mode file content:

The list-mode file generated shows examples of various packets with fields being incremented or decremented. Three types of tag packets are used – i.e. elapsed time, horizontal bed position, and lost event tally. [The elapsed time and lost event tally packets are incrementing except for the last maximum value packets. The horiz. bed position packets are decrementing except for the last, most-negative value packet.] Seven types of event packet fields are shown, all incrementing individually – i.e. AX, BX, XE, AY, BY, TOF, Prompt.

Appendix 1: Here is the content of the list-mode generating C-code file, ml_ea4r_64dp_walkthrough_1.c:

```
//file: ml_ea4r_64dp_walkthrough_1.c
// make list-mode file - software list mode generator
// Eagle 4-Ring 64-bit detector-pair packet field walkthrough
// 8-Oct-2012 wfj

#include <stdio.h>
#include <ctype.h>
#include <math.h>
#include <stdlib.h>
#include <iostream.h>

FILE *streamt;

main (int argc, char **argv)
{
    static int i,j;
    static int e1, e2;
    static int b1, b2;
    static int let1, let2;
    static unsigned __int64 il;

    static char *out_file;

    static int qb;
    static int ax, bx;      // 0 to 51
    static int ay, by;      // 0 to 51 for 4 rings
    static int xe;          // 1 to 54
    static int prompt;      // 0 to 1
    static int tof;          // -24 to 27
    static unsigned int ew1, ew2;
    static unsigned int ms = 0;      // milliseconds
    static int hbp = 0;      // horizontal bed position
    static unsigned int lost = 0; // lost event tally quantity
```

```

// get a filename from the command line
if (argc < 2)
{
    printf ("usage: %s out_file_name\n",argv[0]);
    exit(1);
}
else
    out_file = argv[1];

    streamt = fopen (out_file, "wb");
    if ( streamt == NULL) {
        printf ("No file opened %s\n",out_file);
        exit (1);
    }

// Load Output File

// Incrementing Elapased Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

// Packets to Step through ax Field
ax = 0; bx = 0; xe = 1;      ay = 0; by = 0; prompt = 1; tof = 0;
for (ax=0; ax <= 51; ax++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Incrementing Elapased Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

```

```

// Packets to Step through bx Field
ax = 0; bx = 0; xe = 1;          ay = 0; by = 0; prompt = 1; tof = 0;
for (bx=0; bx <= 51; bx++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Incrementing Elapsed Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

// Packets to Step through xe Field
ax = 0; bx = 0; xe = 1;          ay = 0; by = 0; prompt = 1; tof = 0;
for (xe=1; xe <= 54; xe++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Incrementing Elapsed Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

// Packets to Step through ay Field
ax = 0; bx = 0; xe = 1;          ay = 0; by = 0; prompt = 1; tof = 0;
for (ay=0; ay <= 51; ay++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

```

```

// Incrementing Elapsed Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

// Packets to Step through by Field
ax = 0; bx = 0; xe = 1;      ay = 0; by = 0; prompt = 1; tof = 0;
for (by=0; by <= 51; by++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Incrementing Elapsed Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
let1 = 0x40000000 | (lost          & 0xffff);
let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
j = fwrite (&let1, sizeof(qb), 1, streamt);
j = fwrite (&let2, sizeof(qb), 1, streamt);

// Packets to Step through tof Field
ax = 0; bx = 0; xe = 1;      ay = 0; by = 0; prompt = 1; tof = 0;
for (tof=-24; tof <= 27; tof++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Incrementing Elapsed Time Tag Packet
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms++ >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Decrementing Horizontal Bed Position Tag Packet
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp-- >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

```

```

// Incrementing Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
    let1 = 0x40000000 | (lost          & 0xffff);
    let2 = 0x8000bc00 | ((lost++ >> 16) & 0xf);
    j = fwrite (&let1, sizeof(qb), 1, streamt);
    j = fwrite (&let2, sizeof(qb), 1, streamt);

// Example of One Delayed & One Prompt Events
ax = 0; bx = 0; xe = 1;      ay = 0; by = 0; prompt = 1; tof = 0;
for (prompt=0; prompt <= 1; prompt++) {
    ew1 = ax | (ay<<8) | ((xe&0x7)<<16) | ((tof & 0x7)<<25);
    ew2 = bx | (by<<8) | ((xe>>3)<<16) | (((tof >> 3) & 0x7)<<25) |
((prompt&1)<<30) | (1<<31);
    j = fwrite (&ew1, sizeof(qb), 1, streamt);
    j = fwrite (&ew2, sizeof(qb), 1, streamt);
    il = il + 1;
}

// Maximum Value for Elapsed Time Tag Packet
ms = 0x1fffffff; // all bits in 29-bit field set to one
e1 = 0x40000000 | (ms          & 0xffff);
e2 = 0x80008000 | ((ms >> 16) & 0x3fff);
j = fwrite (&e1, sizeof(qb), 1, streamt);
j = fwrite (&e2, sizeof(qb), 1, streamt);

// Minimum Value (most negative) Horizontal Bed Position Tag Packet
hbp = 0x80000;
b1 = 0x40000000 | (hbp          & 0xffff);
b2 = 0x8000c400 | ((hbp >> 16) & 0xf);
j = fwrite (&b1, sizeof(qb), 1, streamt);
j = fwrite (&b2, sizeof(qb), 1, streamt);

// Maximum Value Lost Event Tally Tag Packet - Type 7 (GIM) & NonFunctional as a Loss
Tally
    lost = 0xfffff; // all bits in 20-bit field set to one
    let1 = 0x40000000 | (lost          & 0xffff);
    let2 = 0x8000bc00 | ((lost >> 16) & 0xf);
    j = fwrite (&let1, sizeof(qb), 1, streamt);
    j = fwrite (&let2, sizeof(qb), 1, streamt);

    printf (" number of 64-bit packets output: %I64d file size: %I64x\n",il,8*il);
    fclose (streamt);
    exit(0);
}

```

Appendix 2: Here is the content of the list-mode verbose reporting C-code file, lmsw64dp_verbose_1.c:

```
// file: lmsw64dp_verbose_1.c
// Verbose Listing for 64 Bit Detector-Pair Packets

// 9-Oct-2012 wfj

#include <stdio.h>
#include <ctype.h>
#include <math.h>
#include <stdlib.h>
#include <iostream.h>

FILE *streami;

main (int argc, char **argv)
{
    static __int64 i,j;
    static __int64 i64_il;

    static char *in_file;

    static int qb;
    static int ax, bx, xe, ay, by, tof, prompt;
    static unsigned int ewl, ew2;
    static int sync;
    static int tag;

    static int ms;           // milliseconds
    static int hbp;          // horizontal bed position value
    static int hbp_se;       // sign-extended horiz. bed position value
    static int lost;         // lost event tally value
    static int tof_se;       // sign-extended tof value

    // get a filename from the command line

    if (argc < 1) {
        printf ("usage: %s in_file_name\n",argv[0]);
        exit(1);
    }
    else { in_file = argv[1];}

    streami = fopen (in_file, "rb");
    if ( streami == NULL) {
        printf ("No file opened %s\n",in_file);
        exit (1);
    }

    i64_il = 0;
```

```

while ((i = fread (&ew1, sizeof(qb), 1, streami) ) != 0) {

    if ((j = fread (&ew2, sizeof(qb), 1, streami) ) != 0) {

        sync = (!((ew1>>31) & 1)) && ((ew2>>31) & 1);

        if (!sync) {
            printf(" PACKET SYNC ERROR 64_bit_word_cnt: %I64d ew1 ew2: %x\n", i64_il, ew1, ew2);
            exit (1);
        }

        if (sync) {

            i64_il = i64_il + 1;

            tag = ((ew1>>28)&0x4)!=0;

            ax = bx = xe = ay = by = tof = prompt = 0;

            if(!tag) {
                ax = ew1 & 0xff;
                bx = ew2 & 0xff;
                xe = ((ew1>>16) & 7) | (((ew2>>16) & 7)<<3);
                ay = (ew1>>8) & 0xff;
                by = (ew2>>8) & 0xff;
                tof = ((ew1>>25) & 7) | (((ew2>>25) & 7) << 3);
                tof_se = tof;
                if (((tof >> 5) & 1) == 1) tof_se = tof | 0xffffffc0;
                prompt = (ew2>>30) & 1;

                printf(" EVENT: pkt_cnt: %I64d ew2 ew1(h): %8x %8x ax bx:
%3d %3d xe %2d ay by: %3d %3d tof(h): %2x tof_se(d): %3d prompt: %1d\n",
                    i64_il, ew2, ew1,
                    ax, bx, xe, ay, by, tof, tof_se,
                    prompt);
            }

            if ( tag){
                if ( ((ew2 >> 12) & 0xe ) == 0x8 ) { // Elapsed Time Tag
                    Packet
                        ms = ( ((ew2 & 0x1fff)<<16) | (ew1 & 0xffff) ); //
                    Extract 29-bit millisecond field
                        printf(" TAG64_ElapsedTime:      pkt_cnt: %I64d ew2
ew1(h): %8x %8x ms(h): %7x ms(d): %9d\n",
                            i64_il, ew2, ew1, ms, ms);
                }
                if ( ((ew2 >> 8) & 0xff) == 0xc4) { // Horizontal Bed
                    Position Tag Packet
                        hbp = ( ((ew2 & 0xff)<<16) | (ew1 & 0xffff) ); //
                    Extract 20-bit bed position field
                        hbp_se = hbp; // Assume hbp is Zero
                        if (((hbp >> 19) & 1) == 1) hbp_se = hbp |
0xffff00000; // Need to Sign Extend
                        printf(" TAG64_HorizBedPos:      pkt_cnt: %I64d ew2
ew1(h): %8x %8x hbp(h): %6x hbp_se(d): %8d\n",
                            i64_il, ew2, ew1, hbp, hbp_se);
                }
            }
        }
    }
}

```

```

        if ( ((ew2 >> 8) & 0xfc) == 0xbc) { // Lost Event Tally
Tag Packet - Type 7 (GIM)
        lost = ( ((ew2 & 0xff)<<16) | (ew1 & 0xffff) ); //
Extract 20-bit lost tally field
        printf(" TAG64_LostEventTally: pkt_cnt: %I64d ew2
ew1(h): %8x %8x lost(d): %7d\n",
i64_i1,          ew2,ew1,          lost);
        }
    }
}

fclose (stream1);
exit(0);
}

```


Appendix 3: Here is an abbreviation of the text report generated by that “verbose” C-code for the example *.l64 file:

```

TAG64_ElapsedTime:    pkt_cnt: 1 ew2 ewl(h): 80008000 40000000 ms(h):          0 ms(d):
0
  TAG64_HorizBedPos:    pkt_cnt: 2 ew2 ewl(h): 8000c400 40000000 hbp(h):          0
hbp_se(d):            0
  TAG64_LostEventTally: pkt_cnt: 3 ew2 ewl(h): 8000bc00 40000000 lost(d):          0
  EVENT: pkt_cnt: 4 ew2 ewl(h): c0000000    10000 ax bx:    0  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 5 ew2 ewl(h): c0000000    10001 ax bx:    1  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  ...
  EVENT: pkt_cnt: 54 ew2 ewl(h): c0000000    10032 ax bx:    50  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 55 ew2 ewl(h): c0000000    10033 ax bx:    51  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  TAG64_ElapsedTime:    pkt_cnt: 56 ew2 ewl(h): 80008000 40000001 ms(h):          1 ms(d):
1
  TAG64_HorizBedPos:    pkt_cnt: 57 ew2 ewl(h): 8000c40f 4000ffff hbp(h):    fffff
hbp_se(d):            -1
  TAG64_LostEventTally: pkt_cnt: 58 ew2 ewl(h): 8000bc00 40000001 lost(d):          1
  EVENT: pkt_cnt: 59 ew2 ewl(h): c0000000    10000 ax bx:    0  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 60 ew2 ewl(h): c0000001    10000 ax bx:    0  1 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  ...
  EVENT: pkt_cnt: 109 ew2 ewl(h): c0000032    10000 ax bx:    0  50 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 110 ew2 ewl(h): c0000033    10000 ax bx:    0  51 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  TAG64_ElapsedTime:    pkt_cnt: 111 ew2 ewl(h): 80008000 40000002 ms(h):          2 ms(d):
2
  TAG64_HorizBedPos:    pkt_cnt: 112 ew2 ewl(h): 8000c40f 4000fffe hbp(h):    ffffe
hbp_se(d):            -2
  TAG64_LostEventTally: pkt_cnt: 113 ew2 ewl(h): 8000bc00 40000002 lost(d):          2
  EVENT: pkt_cnt: 114 ew2 ewl(h): c0000000    10000 ax bx:    0  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 115 ew2 ewl(h): c0000000    20000 ax bx:    0  0 xe 2 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  ...
  EVENT: pkt_cnt: 166 ew2 ewl(h): c0060000    50000 ax bx:    0  0 xe 53 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 167 ew2 ewl(h): c0060000    60000 ax bx:    0  0 xe 54 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  TAG64_ElapsedTime:    pkt_cnt: 168 ew2 ewl(h): 80008000 40000003 ms(h):          3 ms(d):
3
  TAG64_HorizBedPos:    pkt_cnt: 169 ew2 ewl(h): 8000c40f 4000fffd hbp(h):    ffffd
hbp_se(d):            -3
  TAG64_LostEventTally: pkt_cnt: 170 ew2 ewl(h): 8000bc00 40000003 lost(d):          3
  EVENT: pkt_cnt: 171 ew2 ewl(h): c0000000    10000 ax bx:    0  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 172 ew2 ewl(h): c0000000    10100 ax bx:    0  0 xe 1 ay by:    1  0
tof(h): 0 tof_se(d):    0 prompt: 1
  ...
  EVENT: pkt_cnt: 221 ew2 ewl(h): c0000000    13200 ax bx:    0  0 xe 1 ay by:    50  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 222 ew2 ewl(h): c0000000    13300 ax bx:    0  0 xe 1 ay by:    51  0
tof(h): 0 tof_se(d):    0 prompt: 1
  TAG64_ElapsedTime:    pkt_cnt: 223 ew2 ewl(h): 80008000 40000004 ms(h):          4 ms(d):
4
  TAG64_HorizBedPos:    pkt_cnt: 224 ew2 ewl(h): 8000c40f 4000fffc hbp(h):    ffffc
hbp_se(d):            -4
  TAG64_LostEventTally: pkt_cnt: 225 ew2 ewl(h): 8000bc00 40000004 lost(d):          4
  EVENT: pkt_cnt: 226 ew2 ewl(h): c0000000    10000 ax bx:    0  0 xe 1 ay by:    0  0
tof(h): 0 tof_se(d):    0 prompt: 1
  EVENT: pkt_cnt: 227 ew2 ewl(h): c0000100    10000 ax bx:    0  0 xe 1 ay by:    0  1
tof(h): 0 tof_se(d):    0 prompt: 1
  ...
```

```

EVENT: pkt_cnt: 276 ew2 ew1(h): c0003200 10000 ax bx: 0 0 xe 1 ay by: 0 50
tof(h): 0 tof_se(d): 0 prompt: 1
EVENT: pkt_cnt: 277 ew2 ew1(h): c0003300 10000 ax bx: 0 0 xe 1 ay by: 0 51
tof(h): 0 tof_se(d): 0 prompt: 1
TAG64_ElapsedTime: pkt_cnt: 278 ew2 ew1(h): 80008000 40000005 ms(h): 5 ms(d):
5
TAG64_HorizBedPos: pkt_cnt: 279 ew2 ew1(h): 8000c40f 4000fffb hbp(h): ffffb
hbp_se(d): -5
TAG64_LostEventTally: pkt_cnt: 280 ew2 ew1(h): 8000bc00 40000005 lost(d): 5
EVENT: pkt_cnt: 281 ew2 ew1(h): ca000000 10000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 28 tof_se(d): -24 prompt: 1
EVENT: pkt_cnt: 282 ew2 ew1(h): ca000000 2010000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 29 tof_se(d): -23 prompt: 1
...
EVENT: pkt_cnt: 331 ew2 ew1(h): c6000000 4010000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 1a tof_se(d): 26 prompt: 1
EVENT: pkt_cnt: 332 ew2 ew1(h): c6000000 6010000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 1b tof_se(d): 27 prompt: 1
TAG64_ElapsedTime: pkt_cnt: 333 ew2 ew1(h): 80008000 40000006 ms(h): 6 ms(d):
6
TAG64_HorizBedPos: pkt_cnt: 334 ew2 ew1(h): 8000c40f 4000fffa hbp(h): ffffa
hbp_se(d): -6
TAG64_LostEventTally: pkt_cnt: 335 ew2 ew1(h): 8000bc00 40000006 lost(d): 6
EVENT: pkt_cnt: 336 ew2 ew1(h): 80000000 10000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 0 tof_se(d): 0 prompt: 0
EVENT: pkt_cnt: 337 ew2 ew1(h): c0000000 10000 ax bx: 0 0 xe 1 ay by: 0 0
tof(h): 0 tof_se(d): 0 prompt: 1
TAG64_ElapsedTime: pkt_cnt: 338 ew2 ew1(h): 80009fff 4000ffff ms(h): 1fffffff ms(d):
536870911
TAG64_HorizBedPos: pkt_cnt: 339 ew2 ew1(h): 8000c408 40000000 hbp(h): 80000
hbp_se(d): -524288
TAG64_LostEventTally: pkt_cnt: 340 ew2 ew1(h): 8000bc0f 4000ffff lost(d): 1048575

```

Appendix 4: Here are examples of what the V file viewer utility can show of the *.l64 file:

Getting a copy of Fileviewer.exe: <http://www.fileviewer.com/>

Setting up Fileviewer (Version 12) for 32-bit half-packet displayed per line:

View>>Hex Mode (selected)
View>>Hex Formats>> Dword (selected)
View>>Hex Formats>> Flip Ends (selected)
View>>Hex Formats>> Set Hex Line Length (4)

Start of file as displayed:

03020100	
00000000	40000000
00000004	80008000
00000008	40000000
0000000C	8000C400
00000010	40000000
00000014	8000BC00
00000018	00010000
0000001C	C0000000
00000020	00010001
00000024	C0000000

End of file as displayed:

00000A78	00010000
00000A7C	80000000
00000A80	00010000
00000A84	C0000000
00000A88	4000FFFF
00000A8C	80009FFF
00000A90	40000000
00000A94	8000C408
00000A98	4000FFFF
00000A9C	8000BC0F

Setting up Fileviewer (Version 12) for 64-bit packet displayed per line:

View>>Hex Mode (selected)
View>>Hex Formats>> Double Dword (selected)
View>>Hex Formats>> Flip Ends (selected)
View>>Hex Formats>> Set Hex Line Length (8)

Start of file as displayed:

0706050403020100	
00000000	8000800040000000
00000008	8000C40040000000
00000010	8000BC0040000000
00000018	C000000000010000
00000020	C000000000010001

End of file as displayed:

00000A78	8000000000010000
00000A80	C000000000010000
00000A88	80009FFF4000FFFF
00000A90	8000C40840000000
00000A98	8000BC0F4000FFFF