DAQ Raw Data Format
rev 2.1

ed. M. LeVine

3 June 1999

Transportability

In order to allow for differing byte order on different platforms, within DAQ as well as in the analysis phase, the logical record header must contain enough information so that it can be read by any host. To accomplish this the logical record header is defined to be big-endian.

In the following format, the end-edness of a bank is chosen as is appropriate. In many cases, this will be the end-edness of the producer, since our real-time efficiency is more valuable than the offline reconstruction efficiency. But in some cases, the consumer's CPU cycles are more valuable than those of the producer. Case in point, production of clusters to feed to SL3. These are produced by 18 CPUs in parallel, and consumed by a single SL3, so the end-edness of the SL3 should dominate.

Bank headers as well as **bank contents** are written in the native byte order of the producer (see Bank Headers).

Version numbers

There has to be a mechanism to ensure that the software which unpacks an event is compatible with the packing software. So we need to include some kind of versioning information embedded in the event, in several places, as appropriate.

The format version in the logical record header should refer to the format of the bank headers only.

The format version in the detector-specific banks is independent of the format version in the LRHD. [See Bank Header]

Bank order

The only constraint on bank order is that the *first bank at any level* must be the pointer bank. Its contents point to the remaining banks at the same level, making the remaining banks at this level independent of their position.

Cascaded event building

The most efficient event building requires that no stage in the process needs to invade the pieces contributed by the next lower step in the food chain, to modify or even to read. The cascaded structures, which are never modified by an entity later in the chain, are only encapsulated (headers or index banks added). For example, each i960 packs its data into a self-contained package. This package might contain a number of sub-packages, one for each padrow. Three of these might be glued together, either by the monarch on the rvcr

board or by the SB, into an RB package, and finally, the SB's contribution is an index bank allowing navigation among the various RB contributions.

Another thing to keep in mind is that the EVB is not the only consumer of some of this information - for the cluster output, the SL3 needs to be able to navigate it as well.

Data integrity

We need to have a mechanism to monitor data integrity. For this purpose a CRC is proposed, which is generated for each bank separately, in a fixed fashion, in order to determine whether the offsets and/or data it contains has been corrupted. Generation of this CRC will be turned on for diagnostic purposes.

The following data format is meant to deal with these issues.

Numbering convention

All sequences will start with the number '1' (not '0' as in the C language) for better compatibility with the convention used in offline software. For example, sectors, mezzanine cards, pad rows and pads will be numbered starting with '1'.

Data format

The data format is hierarchical. At the topmost level, there are two entities:

Volume Header

This is found at the beginning of the first physical record of the tape. It is the only entity on the first record. Its contents are to be defined. Its most essential function is to contain any information relating to the medium or device on which it is written which may be necessary for it to be read correctly. The **length** of the Volume Header is **fixed** at 4096 bytes.

Logical Record

This can be one of several types: Begin Run, End Run, Data, Slow Control. Each logical record begins with a Logical Record Header, followed by one or more banks. The banks, in most cases contain offsets pointing to other banks.

A logical record is the complete set of data words written in a single write transaction to tape following the acceptance of 1 or more triggers [see *blocking factor*, below]. A logical record will always begin on a block boundary. The length of a logical record will always be a multiple of 4 bytes.

Begin Run: this will signal the beginning of a run.
Contents: to be defined.

End Run: this signals the end of a run.
Contents: to be defined.

Data: This entity is the most frequent to be found on the tape. It

contains one or more data events, containing banks for the major detector components.

Slow Control: this record contains slow control information.
Contents: to be defined.

Logical Record Header (10 words)

The Logical Record Header is always at the beginning of a physical record.

Words 1-2: Bank type
"LRHD "
These bytes (8 ASCII characters padded to 8 bytes by trailing blanks) specify the bank type

Word 3: Length of bank (words) including the 10 word bank header.
15 (5 in logical record body + 10 in header).

Word 4: Run number

Word 5: Format version
65536*major + minor version number
This format version applies to all BANK HEADERS in the record. It does not apply to the contents of each bank, which are specified separately for each bank.

Word 6: Byte order
This word will always be written as **0x04030201** by the producer. If it reads as 0x04030201, the producer's byte order is the same as that of the reader; if not, bytes need to be swapped.

Word 7: reserved

Word 8: reserved

Word 9: Reserved

Word 10: CRC
Calculated 32 bits wide for this bank only, including the bank header excluding the word containing the CRC. Byte ordering of **the producer** is used. The CRC generation is turned on by software command during bank creation. If CRC generation is disabled, it will be written as 0.

Logical Record body:

Word 1: length
This is the length, in words, from word 1 of the Logical Record header to the end of the record, i.e., including all banks that belong to the record. This is to allow easy navigation past a given record.

Word 2: blocking factor

This is the number of events contained in this Logical Record. It will be =1 for large events, but allows us to avoid excessive overhead for small events (e.g., EMC only).

Words 3-4: Record type (right padded with spaces to 8 characters)

'BEGR '
'ENDR '
'DATA '
'SLOW '

Word 5: CRC

Calculated 32 bits wide for the entire **payload** of the logical record. Calculation of the CRC begins with the word following the Logical Record Bank, includes all headers and contents of every bank in the logical record. Byte ordering of **the producer** is used. [See *bank headers*, below.]

Bank headers (10 words)

The *byte order* of the bank header must be the same as the byte order of the bank contents. In general the byte order will be the native byte order of the producer. A consumer needs to examine Word 6 to determine whether (and how) the bank header and bank contents need to be byte-swapped.

The *format version* must be identical for all banks of a given detector. If the offline software detects inconsistencies in the format version among the various banks of a detector, it is expected to stop processing the event.

Words 1-2: Bank type

These bytes (up to 8 ASCII characters padded, if necessary, to 8 bytes by trailing blanks) specify the bank type

Word 3: Length of bank (words) including the 10 word bank header.

The bank length should not be used to deduce the location of the next bank, as gaps may exist between banks. The length and offset entries in the next higher pointer bank should be used for this purpose.

Word 4: Bank identifier. Used to distinguish between multiple banks of the same type produced in a given context.

Word 5: Format version

Contains major and minor format version encoded as in logical record header

Word 6: Byte order

This word will always be written as **0x04030201** by the producer. If it

reads as 0x04030201, the producer's byte order is the same as that of the reader; if not, bytes need to be swapped.

Word 7: Format number
defined for each bank separately

Word 8: Token number

Word 9: Reserved

Word 10: CRC
Calculated 32 bits wide for this bank only, including the bank header excluding the word containing the CRC. Big endian byte ordering is used. The CRC generation is turned on by software command during bank creation. If CRC generation is disabled, it will be written as 0.

Naming convention:

Bank types consist of up to eight letters or numbers, padded, if necessary, with trailing blanks. The *first* 3 letters specify the *subdetector originating* the bank:

- TPC TPC
- EMC EMC
- SMX Shower Max
- SVT SVT
- FTP FTPC
- TOF TOF
- L3_ Level 3
- TRG Trigger

The *intermediate* 0 to 4 letters, if present, signify the organizational unit or payload, which is not necessarily unique outside of the immediate context, e.g.:

- SEC Sector
- RB Receiver board
- MZ Mezzanine board
- SEQ Sequence
- ADC ADC data
- CPP Cluster pointer
- CFG Configuration data
- PAD TPC pad
- PR TPC pad row
- BAD Bad channel
- PED Pedestal
- RMS pedestal RMS
- GLB Global (L3)
- GAIN Gain

The *last* letter signify the *type* of bank:

- P Pointer Bank. A pointer bank carries information as to the existence of lower-level banks, their lengths, and their positions relative to the pointer bank.
- X Index Bank. The index bank carries information to assist in locating data from, e.g., a pad row, in a bank containing zero suppressed data.
- K Key Bank. A key bank carries information on the pad/pad row combination sequence in a **raw bank**.
- D Data bank. The data bank carries data or auxiliary data, associated with *logical* pads (cf. **raw bank** below).
- R Raw bank. A raw bank carries information relating to the physical pads associated with a mezzanine board. The information in a raw bank corresponds to the =384 pads physically mapped to a mezzanine, as contrasted to the logical pads associated with the mezzanine.
  Logical pads may be *borrowed* from adjacent mezzanines in order to arrive at a more compact representation (less isolated pads in a pad row).

Bank types:

Some bank types are listed below, where those labeled with an asterisk are detailed in this document. [The first 5 listed banks do not conform to the naming convention described above.]

BEGRUN*
PEDESTAL
CALIB
ENDRUN*
SLOWCTL
DATAP*
L3_P
L3_SECLP
L3_SECLD
TPCP*
TPCSECP*
TPCRBP*
TPCMZP*
TPCADCD*
TPCADCR*
TPCPEDR*
TPCGAINR*
TPCBADR*
TPCRMSR

TPCPRP*
TPCSEQX*
TPCSEQD*
TPCRBP*
TPCMZP*
TPCMZCLD*

BEGRUN

Length in this header refers to the BEGRUN bank itself. It does not include the length of the events which comprise the run. Contents: TBD

ENDRUN

Length in this header refers to the ENDRUN bank itself. It does not include the length of the events which comprise the run. Contents: TBD

DATAP (Data Pointer Bank)

Length in this header refers to the DATAP bank itself. The contents of word 1 of the DATAP bank contain the length of the entire event; this allows the reader to easily skip over the entire event if desired. This is useful when the blocking factor (see logical record header) >1 and thus multiple DATAP banks are present within a logical record.

This bank MUST be the first bank of each event in the logical record. The first bank of any event must be a Data Pointer Bank, which will contain pointers to the first word of the defining banks for each major component - i.e. TPC, SVT, FTPC, EMC, GL3, TOF etc - that follows, calculated from the first word of the Data Pointer Bank header. The Data Pointer Bank also contains lengths for all detector contributions, to facilitate assembly of a partial event.

Now follow banks for major detectors. These may appear in any order.

There may be banks within an event which may contain critical slow control data such as TPC pressure and temperature monitors or anything else that offline would like to have available on an event by event basis, but not dumps of slow controls data which are unrelated to a specific event.

A bank for a given major detector (e.g. TPC, SVT, FTPC, EMC, GL3) may contain offsets pointing to one or many subsidiary banks containing data from sub-parts of the detector (e.g., a TPC major sector). Thus these subsidiary banks may be in any order (see bank definitions which follow).

The word count (word 3 of the bank header) in all cases contains only the length of the bank itself. The Data Pointer Bank provides sufficient information to navigate among the detector components. All detector contributions begin with a pointer bank with pointers to its sub-components, and this principle is carried out for lower hierarchical structures as well: where a pointer bank is used, it must be the first bank at its level.

The first bank of any detector's contribution, pointed to by the Data Pointer Bank, is the Pointer Bank for that detector.

DATA Type Logical Record Contents:

- DATAP Bank
- TPCP Bank
  - TPCSECP Banks
    - TPCRBP Banks
      - TPCMZP Banks
        - TPCPADK Bank
        - TPCADCR Bank
        - TPCADCX Bank
        - TPCADCD Bank
        - TPCCSEQX Bank
        - TPCSEQD Bank
        - TPCCPPR Bank

DATAP (DATA Pointer Bank)

Bank Contents:

Data word 1:	Length of following event, in words, from first header word of DATAP bank.
Data word 2:	Time (Unix format)
Data word 3:	Event sequence number (unique within a run, not necessarily consecutive).
Data word 4:	Trigger word
Data word 5:	Trigger input word
Data word 6:	Detector presence bits: 0. TPC 1. SVT 2. TOF 3. EMC 4. Shower Max 5. FTPC 6. Reserved 7. RICH 8. Trigger detectors 9. L3
Data words 7-26:	Offset, in words, from first word of Data Pointer bank header to the detector bank, and length (words) of each contribution: TPC offset TPC length

SVT offset
SVT length
TOF offset
TOF length
EMC offset
EMC length
Shower Max offset
Shower Max offset
FTPC offset
FTPC length
RICH offset
RICH length
reserved 3
reserved 4
TRG offset
TRG length
L3 offset
L3 length

Data word 27-128: reserved

Detector components (position as indicated in the Data Pointer Bank, words 7-26):

TPCP (TPC Pointer Bank)

Format version: 2.0
This short bank indicates the relative position and length of each TPC double sector bank. Note that the TPC data do not contain clusters (space points) except for debugging purposes. These are found under GL3 under normal circumstances.

There are 24 sectors provided for in this bank. During year 1, only the twelve odd sector entries will be non-zero. [This transitional form is denoted by the header word **Format number** =1.]

Length is 58 (48 data words).
Contents:

Data words 1-48: Offset (words) from first word of the TPC contribution to sector contribution for each sector, followed by length (words) of each contribution. A zero signifies no contribution is present.
Format number=1: (Year 1) A double sector contribution is indicated by each offset and length. Only odd sector numbers are represented, each corresponding to a double sector (12 receiver boards).
Format number>1: Each entry corresponds to a single sector (6 receiver boards).

Sector 1 offset
Sector 1 length
Sector 2 offset
Sector 2 length [=0 for Format number = 1]
Sector 3 offset
Sector 3 length
Sector 4 offset
Sector 4 length [=0 for Format number = 1]

etc.

TPCSECP (TPC Sector Pointer Bank)

This bank allows navigation among the banks containing the contributions from the various receiver boards corresponding to this sector. The bank identification (1,2,3,…,24) indicates the sector. [Note that during year 1 there will be only odd sectors present, and the **Format number** for TPCSECP (word 7 of the header) will be 1.]
Length in this header is 34 (24 data words).

Data words 1-24: Offset (words) from first word of TPCSECP header to the sector contribution for receiver boards 1,...,12. Receiver boards 1-6 correspond to the odd sector, 7-12 to the even sector. Thus, sector 1 maps to RBs 1-6, sector 2 maps to RBs 7-12. A zero signifies no contribution is present., e.g.,

Word 1: Offset to receiver board 1 contribution
Word 2: Length of receiver board 1 contribution
Word 3: Offset to receiver board 2 contribution
Word 4: Length of receiver board 2 contribution
etc.

Format number = 1: (Year 1) Only odd sector banks will be present, each with 12 receiver boards. Receiver boards 1-6 correspond to the sector whose number appears in the Bank ID word, while boards 7-12 correspond to the sector whose number is one greater than the Bank ID.
Format number > 1: (After Year 1) All 24 sector banks will be present, each with 6 receiver boards.

TPCRBP (TPC Receiver Board Pointer Bank)

Ability to skip among sector contributions is provided by the next higher pointer bank (TPCSECP). The bank identification in the header corresponds to the receiver board number (1-12).
Length in this header is 32 (22 data words).

Data word 1: Offset in words to the mezzanine board A pointer bank

Data word 2:	Length in words of the mezzanine board A pointer bank
Data word 3:	Offset in words to the mezzanine board B pointer bank
Data word 4:	Length in words of the mezzanine board B pointer bank
Data word 5:	Offset in words to the mezzanine board C pointer bank
Data word 6:	Length in words of the mezzanine board C pointer bank [Offset of 0 signifies a bank is not present]
Data wds 7-22:	Header as received via the fiber (64 bytes)

TPCMZP (TPC Mezzanine Board Pointer Bank)

The bank identification in the header corresponds to the mezzanine board number (1-3). **Length** in this header is **variable**; unused offset/length pairs may be omitted from *the end of the list*.

Data word 1:	Offset in words to the TPCADCD bank (zero-suppressed ADC data)
Data word 2:	Length in words of the TPCADCD bank
Data word 3:	Offset in words to the TPCSEQD bank (sequence data)
Data word 4:	Length in words of the TPCSEQD bank
Data word 5:	Offset in words to the TPCADCX bank (index to ADC data and sequences)
Data word 6:	Length in words of the TPCADCX bank
Data word 7:	Offset in words to the TPCPADK bank (key to raw ADC, pedestal, RMS, configuration, and gain data)
Data word 8:	Length in words of the TPCPADK bank
Data word 9:	Offset in words to the TPCCPPR bank (raw cluster pointer pairs)
Data word 10:	Length in words of the TPCCPPR bank
Data word 11:	Offset in words to the TPCADCR bank (unsuppressed (raw) ADC data)
Data word 12:	Length in words of the TPCADCR bank
Data word 13:	Offset in words to the TPCMZCLD bank (cluster data - normally exported by SL3)
Data word 14:	Length in words of the TPCMZCLD bank
Data word 15:	Offset in words to the TPCCFGR bank (raw configuration data)
Data word 16:	Length in words of the TPCCFGR bank

Data word 17:	Offset in words to the TPCPEDR bank (raw pedestal data)
Data word 18:	Length in words of the TPCPEDR bank
Data word 19:	Offset in words to the TPCRMSR bank (raw pedestal RMS data)
Data word 20:	Length in words of the TPCRMSR bank
Data word 21:	Offset in words to the TPCGAINR bank (raw gain data)
Data word 22:	Length in words of the TPCGAINR bank
Data word 23:	Offset in words to the TPCBADR bank (bad channel list)
Data word 24:	Length in words of the TPCBADR bank

[*Offset of 0 signifies a bank is not present*]

TPCADCD TPC Mezzanine ADC Data Bank

This bank contains only zero-suppressed ADC data.

Bank identification = 1,2,3 for mezzanine A,B,C.

Format number = 0 zero suppressed data, uncompressed
1 zero suppressed data, compressed

Note that for compressed ADC data, this bank must be uncompressed before any of the information contained in TPCADCX, TPCSEQD can be applied to it.

Contents: (format 0)

Words 1...:	ADC data (packed 4 ADC values per word). The last word is padded with trailing zeroes, if required.

Contents: (format 1) Version 1.0

Word 1:	number of encoded ADC values
Words 2-129:	encoding dictionary (one short per entry)
Words 130...:	encoded ADC data

TPCADCR TPC Mezzanine ADC Raw Bank

This bank contains only unsuppressed ADC data.

Bank identification = 1,2,3 for mezzanine A, B, C.

Format number = 0 unsuppressed data, uncompressed
1 unsuppressed data, compressed

Note that for compressed ADC data, this bank must be uncompressed before any of the information contained in TPCPADK can be applied to it.

Contents: (format 0)

Words 1...: ADC data (packed 4 ADC values per word).

Contents: (format 1) Version 1.0

Word 1:	number of encoded ADC values
Words 2-129:	encoding dictionary (one short per entry)
Words 130...:	encoded ADC data

TPCPADK (Mezzanine Pad Key Bank)

This bank serves as a key to the sequences found in the TPCADCR, TPCCPPR, TPCPEDR, TPCRMSR, TPCCFGR, and TPCGAINR banks. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 208 (198 data words).

Data word 1: number of bytes per ADC sequence
Data word 2: number of bytes per CPP sequence
Data word 3: number of bytes per PED sequence
Data word 4: number of bytes per RMS sequence
Data word 5: number of bytes per CFG sequence
Data word 6: number of bytes per GAIN sequence
Data word 7-198:

```
struct {
        struct {
                unsigned char pad row;
                unsigned char pad;
        };
struct {
                unsigned char pad row;
                unsigned char pad;
        };
};
```

(outermost struct repeated 192 times, where invalid pad, pad row combinations are signified by 0xff for both pad and pad row.)

TPCADCX (Mezzanine ADC Index Bank)

This bank is present only in conjunction with TPCADCD and TPCSEQD.

Data word 1:	pad row #
Data word 2:	offset (in bytes) into (uncompressed) TPCADCD bank to beginning of ADC data for this pad row
Data word 3:	offset (in bytes) into TPCSEQD bank to beginning of SEQ data for this pad row
Data words 4 ...	repeat words (1-3) as necessary (up to a total of 6 times)

TPCCPPR (Cluster Pointer Raw Bank)

The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 12299 (12289 data words). This bank contains the raw cluster pointers provided by the ASICs. Note that the least 10 bits are significant, and the remaining bits should all be zero if the sequence is a valid one. Invalid sequences are marked by one or more of the high order bits set to 1. The pad and pad row order are defined in the TPCPPK bank.

Data word 1: ASIC cluster finding parameters

```
struct {
    unsigned char thresh_lo;
    unsigned char thresh_hi,;
    unsigned char n_seq_lo;
    unsigned char n_seq_hi;
};
```

Data word 2-33: start time bin (16 bits), stop time bin (16 bits) [pad 1]
Data word 34-65: start time bin (16 bits), stop time bin (16 bits) [pad 2]
Data word 12289: 32 words per pad, for pads 1-384

TPCSEQD (Mezzanine Sequence Data Bank)

This bank is present only in conjunction with TPCADCD.
Bank identification = mezzanine number

Data word 1-n: sequence (packed 2/word)

bit 15:	Switch (discriminates between 2 formats following)	
Switch = 0	bits 14-6:	start time
	bit 5:	last sequence (this pad)
	bit 4-0:	sequence length
Switch = 1	bits 14-0:	s_pad number = 256*pad row + pad number

If (*switch==0*) first member of union is used; else next 15 bits contain a pad number for the next sequence. For heavily populated pad rows, *switch=0.* The assumption is that every pad has at least one sequence to report. The maximum sequence length which can be accommodated in one unsigned short is 32 time bins. If bit 5 is a '1' this signifies that the current sequence is the last for a given pad. The default is that the next sequence corresponds to the next pad in numerical sequence. If this is not the case, *switch=1* for the next unsigned short, together with the next pad number. If a sequence is longer than 32 time bins, it has to be reported in pieces. [Overhead for Au-Au events: ~33%]. Pad number = 255 ==> ignore (dummy spacer to fill out bank)

The first sequence of the bank **must start** with *switch=1*, as must the first sequence of every pad row.

TPCCFGR (Mezzanine Configuration Raw Bank)

This bank carries information derived from a **configuration special event**. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 106 (96 data words).

```
Data word 1-96:     struct {
                        unsigned char FEE_id;
                        unsigned char FEE_id;
                        unsigned char FEE_id;
                        unsigned char FEE_id;
                    };
```

struct repeated 96 times.

TPCBADR (Mezzanine Bad Channel Bank)

This bank carries information derived from a **configuration special event**. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is **variable**.

```
Data word 1-n:      struct {
                        unsigned char row;
                        unsigned char pad;
                    } badChannel[N] ;
```

struct repeated N times (up to 384).

The last word is padded with trailing zeroes, if required.

TPCPEDR (Mezzanine Pedestal Raw Bank)

This bank carries information derived from a series of **pedestal special events**. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 49163 (49153 data words).

```
Data word 1:        # events used in the calculation
Data word 2-49153:  8 bit pedestal values, packed 4 per word.
                    The last word is padded with trailing zeroes, if required.
```

TPCRMSR (Mezzanine RMS Raw Bank)

This bank carries information derived from a series of **pedestal special events**. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 49163 (49153 data words).

```
Data word 1:        # events used in the calculation
```

Data word 2-49153: 8 bit pedestal RMS values, packed 4 per word
as (RMS<<4). The last word is padded with trailing zeroes, if required.

TPCGAINR (Mezzanine Gain Raw Bank)

This bank carries information derived from a series of **gain special events**. The bank identification corresponds to the mezzanine board number (1-3).
Length in this header is 396 (386 data words).

Data word 1: # events used in the calculation
Data word 2: *mean gain* in absolute ADC counts used in conjunction with the relative gains defined below.
Data word 3-386: struct {
UINT16 (t0<<4);
UINT8 (t0_RMS)<<4;
INT8 (rel gain ñ 1)<<6;
};
Data word 387-643: UINT8 trans_table[1024]; //master table used in ASICs
Data word 644-771: UINT16 exp_table[256]; // inverse of above

GL3 banks

The following banks are produced by GL3. (There are pointer banks which still need to be defined...)

TPCSECLP (TPC Sector Cluster Pointer Bank)

This bank allows navigation among the banks containing the clusters (space points) for the various receiver boards corresponding to this sector. The bank identification (1,3,5,...,23) indicates the sector.
Length in this header is 34 (24 data words).

Data words 1-12: Offset (words) from first word of TPCSECLP header to sector contribution for receiver boards 1,..,12, and length of the sector contribution. Receiver boards 1-6 correspond to the even sector, 7-12 to the odd sector. A zero signifies no contribution is present., e.g.,

Word 1: Offset to sector 1 contribution
Word 2: Length of sector 1 contribution
Word 3: Offset to sector 2 contribution
Word 4: Length of sector 2 contribution
etc.

TPCRBCLP (TPC Receiver Board Cluster Pointer Bank)

Length in this header is 32 (22 data words). The bank identification in the header corresponds to the receiver board number (1-12).

Data word 1: Offset in words to the mezzanine board A clusters
Data word 2: Length in words of the mezzanine board A clusters
Data word 3: Offset in words to the mezzanine board B clusters

Data word 4: Length in words of the mezzanine board B clusters
Data word 5: Offset in words to the mezzanine board C clusters
Data word 6: Length in words of the mezzanine board C clusters
[Offset of 0 signifies no contribution.]
Data wds 7-22: Header as received via the fiber (64 bytes)

TPCMZCLD (TPC Mezzanine Board Cluster Data Bank)

Length in this header includes all clusters representing the contribution of this TPC mezzanine board to the SL3. The bank identification in the header corresponds to the mezzanine board number (1-3).

Data word 1: Number of pad rows present in this bank
Data word 2: Pad row
Data word 3: # clusters this pad row
Data word 4:
```
struct  centroids{
        unsigned short X centroid;   /* units: 1/64 pad */
        unsigned short T centroid;   /* units: 1/64 time bin */
};
```
Data word 5:
```
unsigned short flags {
        Reserved;                    /*  bits 7-15   */
        Centroid quality;            /*  bits 3-6    */
        Saturated ADC;               /*  bit 2       */
        Excessive time bin width;    /*  bit 1       */
        Excessive pad width;         /*  bit 0       */
};
unsigned short total charge;
```

{Repeat of words (4,5) as necessary
Repeat words (2,3),
....}
The last word is padded with trailing zeroes, if required.

Note that contributions corresponding to the same pad row, arising from different mezzanine boards, may show up in different banks, due to the hardware distribution of pads to different mezzanines.

SVTP (SVT Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY SVT GROUP

TOFP (TOF Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY TOF GROUP

EMCP (EMC Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY EMC GROUP

SMDP (Shower Max Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY EMC GROUP

FTPP (FTPC Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY FTPC GROUP

RICP (RICH Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY RICH GROUP

TRGP (TRG Pointer Bank) DOCUMENTATION TO BE SUPPLIED BY TRG GROUP

Change log

Version	Date	Whom	Reason
1.12	05-May-99	MJL	Format version within detector must match
1.13	06-May-99	MJL	Change TPCP definition of odd, even sector mapping to RB numbers
1.14	11-May-99	MJL	Change TPCP, TPCSECP to include 24 sectors, 12 of which are used for year 1 operation Scope of CRC for LRHD changed. CRC byte order changed.
2.0	13-May-99	MJL	ASIC parameters introduced in TPCCPPR to enable synthesis of zero-suppressed banks offline Document has format version specified for each detector top level bank (e.g., TPCP) TPCADCX: Worst case length now corresponds to 6 pad rows
2.01	28-May-99	MJL	Changed word count in TPCCPPR description Removed TPCADCR padding description
2.1	3-Jun-99	MJL	TPCGAINR, TPCCFGR deleted zero padding TPCGAINR added translation tables DATAP added RICH entry, trigger bit 7 RICHP, others: added disclaimer