Outer Tracker
DAQ data format

April 14, 2007

Abstract

This document describes the current data format generated from the TELL1 board for the LHCb Outer Tracker.

LHCb Technical Note

Issue	Final
Revision	1.5.1
Reference	LHCb-2007-040, EDMS 833984
Created	July 31, 2006
Last Modified	April 14, 2007

Prepared by Alex Gong
Tsinghua University, Beijing, P.R. China

Jacopo Nardulli, Antonio Pellegrino
NIKHEF, Amsterdam, The Netherlands

Dirk Wiedner
CERN, Geneva, Switzerland

Mirco Nedos
University of Dortmund, Germany

Contents

List of Tables

List of Figures

1 Document Status Sheet

Document Title: Outer Tracker DAQ data format			
Document Reference Number: LHCb-2007-040, EDMS 833984			
Issue	**Revision**	**Date**	**Reason for change**
Draft	1.0	31.07.2006	first version
	1.1	04.08.2006	BxID added into OT specific header. Event size calculation sections added.
	1.2	09.08.2006	Compatibility and debugging support sections added.
	1.3	09.08.2006	GOL header for all enabled links, even those without hit.
	1.4	11.08.2006	Unused PP will send less information to reduce event size.
Final	1.5	07.02.2007	General revision. Changed to LaTeX.
Final	1.5.1	14.04.2007	Bank header description corrected

2 Multiple Event Packet

Each TELL1 board will generate data in Multiple Event Packets (MEPs) as shown in Fig. 1. Each MEP consists of a MEP header and a number of events, given by the MEP factor (1...32). Each event starts with a MEP sub-header followed by several banks of data. Each bank consists of a bank header and the bank data. The formats of the MEP header, of the MEP sub-header and of the bank header have been defined in Ref. [1] and are for the sake of clarity recapitulated below.

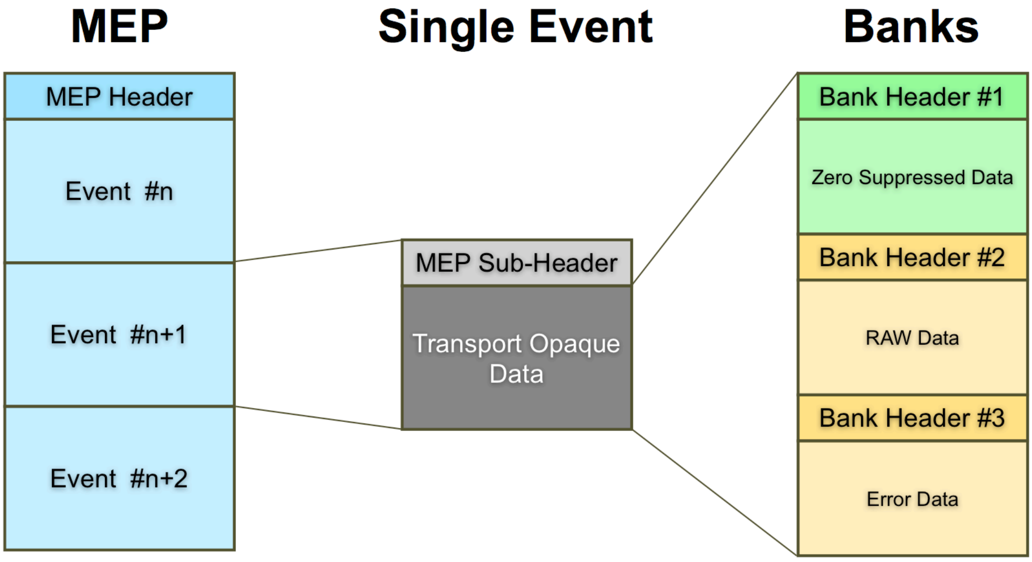


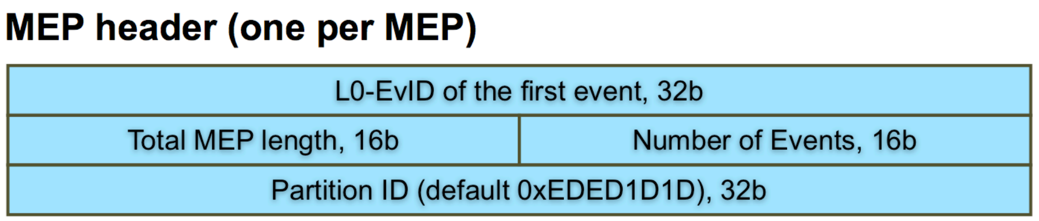
Figure 1: Overview of the MEP hierarchy

2.1 MEP header

The MEP header, consisting of 3 words, 32 bits each, is sent at the begin of each MEP. Its format, defined in Ref. [1], is shown in Fig. 2. Notice that the MEP length field here is the total length (in bytes) of the MEP **including** the 3 word MEP header.

2.2 MEP sub-header

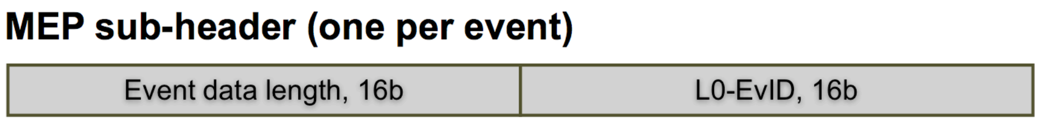
Each event in a MEP contains again a one word (32 bits) MEP sub-header. Its format, defined in Ref. [1], is shown in Fig. 3. The length field here is the length (in bytes) of the event **excluding** the MEP sub-header word.



MEP header (one per MEP)

L0-EvID of the first event, 32b	
Total MEP length, 16b	Number of Events, 16b
Partition ID (default 0xEDED1D1D), 32b	

Figure 2: MEP header



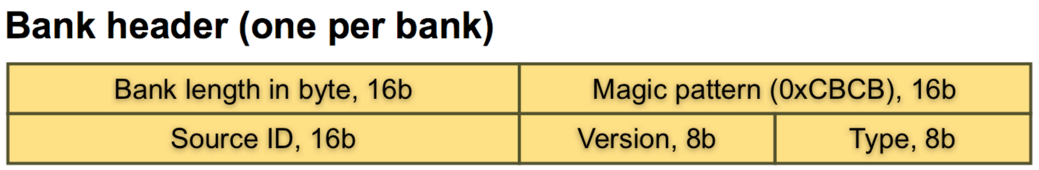
MEP sub-header (one per event)

Event data length, 16b	L0-EvID, 16b

Figure 3: MEP sub-header

2.3 Bank header

Each Data Bank in an Event is preceded by a bank header. Its format, defined in Ref. [1], is shown in Fig. 4.



Bank header (one per bank)

Bank length in byte, 16b	Magic pattern (0xCBCB), 16b	
Source ID, 16b	Version, 8b	Type, 8b

Figure 4: Bank header

- The Bank Length is the length (in bytes) of the bank data **excluding** the padding at the end but **including** the Bank header of eight bytes (2 words). Notice that in zero-suppress process mode the Outer Tracker will have a byte padding for each GOL, that has to be taken into account in the bank length calculation.

- The Source ID will be an identifier of the TELL1 board (i.e. 0x1 to 0x30 for the Outer Tracker).

- The Bank Version gives the software version number for the DAQ (e.g. 0x1 for DC06).

- The Bank Type identifies the content of the bank:

- 12 (0xC): processed bank;
- 32 (0x20): RAW data bank;
- 33 (0x21): Error bank.

Notice that these Bank Type values, the Source ID and the Bank Version number are declared in the TELL1 configuration file.

3 Bank data

Three data banks are foreseen for the Outer Tracker:

- Processed bank. Two processed data formats are foreseen:
  - Zero Suppressed;
  - Hit Maps;
- RAW bank (the original OTIS data);
- Error bank (generated in case of error, providing the details of the failure mode.

While RAW banks (”force_raw_bank” bit in the TELL1 configuration file or dynamically by trigger type 0x5) and Error banks (”force_info_bank” bit in the TELL1 configuration file or automatically if an error occurs) are only generated on request, the Processed Bank always exists in an event, even in the absence of any valid hit in the corresponding part of the Outer Tracker. The sequence of these three banks is fixed:

- Processed Bank - RAW Bank - Error Bank
- Processed Bank - Raw Bank
- Processed Bank - Error Bank
- Processed Bank

3.1 Processed Bank

The format of the Processed Bank is illustrated in Fig. 5. It starts with the header shown in Fig. 4 (Bank Type 0xC identifies a Processed Bank) followed by an “OT specific” header word (32 bits), defined as in Tab. 1. The remaining part of the bank consists of a number (1...24) of GOL data blocks.

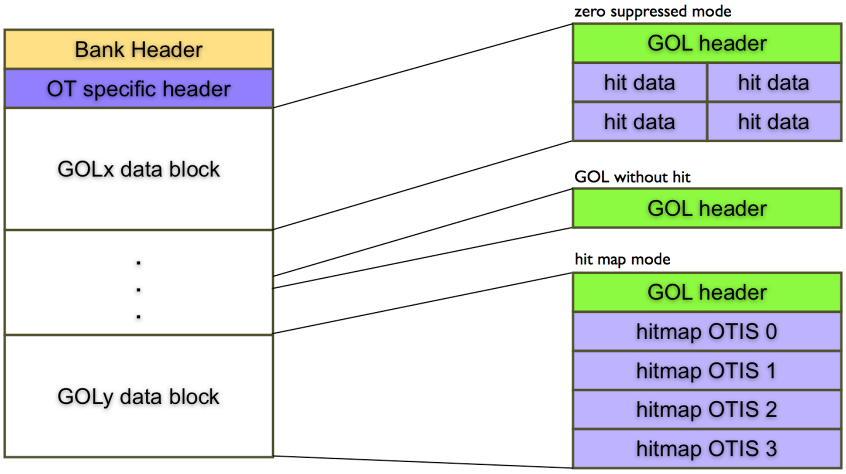


Figure 5: Outer Tracker processed data bank structure

Each GOL data block starts with a GOL header, followed by the actual data. The format of the GOL header, following closely that defined in Ref. [2], is shown in Tab. 2: it contains the total number of hits in the GOL data block, some reduced info from the four OTIS headers, the processed data type (zero suppressed or hitmap), and a GOL ID, uniquely identifying the position of the GOL in the detector according to the address scheme described in Ref. [3] and shown in Tab. 3. All GOL links (enabled in the TELL1 configuration file) will produce a GOL header word, even in the absence of valid hits in the corresponding Front-End Electronics.

The data in the GOL data block can be either in zero-suppress or in hitmap format. The two modes can be configured separately for each link (GOL). Data are generated only if there are valid hits in the corresponding Front-End Electronics: in zero-suppress mode, two bytes per hit are used; the hitmap format has a fixed size of 16 bytes. The format of the processed data follows closely that defined in Ref. [2]. In zero-suppress mode, as shown in Tab. 4, each 16-bits hit word carries a 7-bits straw ID, identifying the OTIS (2 bits) and the channel (5 bits), and an 8-bit drift time; the data is padded to 32-bits words. In hitmap mode, as shown in Tab. 5, no driftime information is available.

Bit(31..28)	not used, all 0
Bit(27..25)	trigger type
Bit(24)	general error flag
Bit(23..16)	bunch counter
Bit(15..0)	number of enabled GOLs

Table 1: **OT specific header**

Bit(31..24)	number of hits in GOL (module)
Bit(23)	optical data transmission ok
Bit(22)	data process mode (1: zero-suppress mode / 0: hitmap mode)
Bit(21..19)	OTIS3 status from OTIS header (SEU, buffer overflow, truncation)
Bit(18..16)	OTIS2 status from OTIS header (SEU, buffer overflow, truncation)
Bit(15..13)	OTIS1 status from OTIS header (SEU, buffer overflow, truncation)
Bit(12..10)	OTIS0 status from OTIS header (SEU, buffer overflow, truncation)
Bit(9..0)	GOL ID

Table 2: **GOL header**

Bit(9..8)	station (01-11)
Bit(7..6)	layer (00-11)
Bit(5..4)	quarter (00-11)
Bit(3..0)	module (0001-1001)

Table 3: **GOL ID in GOL header** [3]

Bit(15)	”1”
Bit(14..13)	OTIS ID
Bit(12..8)	channel number
Bit(7..0)	encoded drift time

Table 4: **GOL data in zero-suppress mode (16 bits per hit)**

word 0, Bit(31..0)	hitmap of OTIS 0; one bit per straw channel (bit(31) corresponds to channel 31, bit(0) to channel 0)
word 1, Bit(31..0)	hitmap of OTIS 1; one bit per straw channel (bit(31) corresponds to channel 31, bit(0) to channel 0)
word 2, Bit(31..0)	hitmap of OTIS 2; one bit per straw channel (bit(31) corresponds to channel 31, bit(0) to channel 0)
word 3, Bit(31..0)	hitmap of OTIS 3; one bit per straw channel (bit(31) corresponds to channel 31, bit(0) to channel 0)

Table 5: **GOL data in hitmap mode (4 words á 32 bits)**

3.2 RAW bank

The format of the RAW Bank is illustrated in Fig. 6. It starts with the bank header shown in Fig. 4 (Bank Type 0x20 identifies a RAW Bank) followed by the data from the **enabled** PreProcessing (PP) FPGAs (PP0–PP3). Each PP FPGA on the TELL1 board can handle up to six optical links. Each PP data block (PPx data) contains the data from the OTIS chips in their integrity (36 bytes per OTIS: 4 bytes for the OTIS header and 32 bytes for the OTIS data), plus an event information block, in the following sequence:

- data from "even" OTIS' (OTIS 0 and OTIS 2) (6 links, 108 words);
- data from "odd" OTIS' (OTIS 1 and OTIS 3) (6 links, 108 words);
- event information section (17 words).

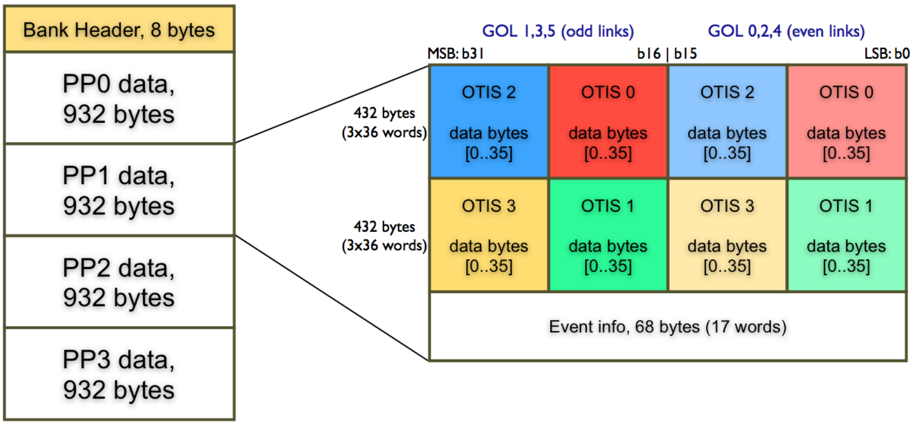

Bank Header, 8 bytes
PP0 data, 932 bytes
PP1 data, 932 bytes
PP2 data, 932 bytes
PP3 data, 932 bytes
GOL 1,3,5 (odd links)
GOL 0,2,4 (even links)
MSB: b31
b16 | b15
LSB: b0
432 bytes (3x36 words)
432 bytes (3x36 words)
OTIS 2 data bytes [0..35]
OTIS 0 data bytes [0..35]
OTIS 2 data bytes [0..35]
OTIS 0 data bytes [0..35]
OTIS 3 data bytes [0..35]
OTIS 1 data bytes [0..35]
OTIS 3 data bytes [0..35]
OTIS 1 data bytes [0..35]
Event info, 68 bytes (17 words)

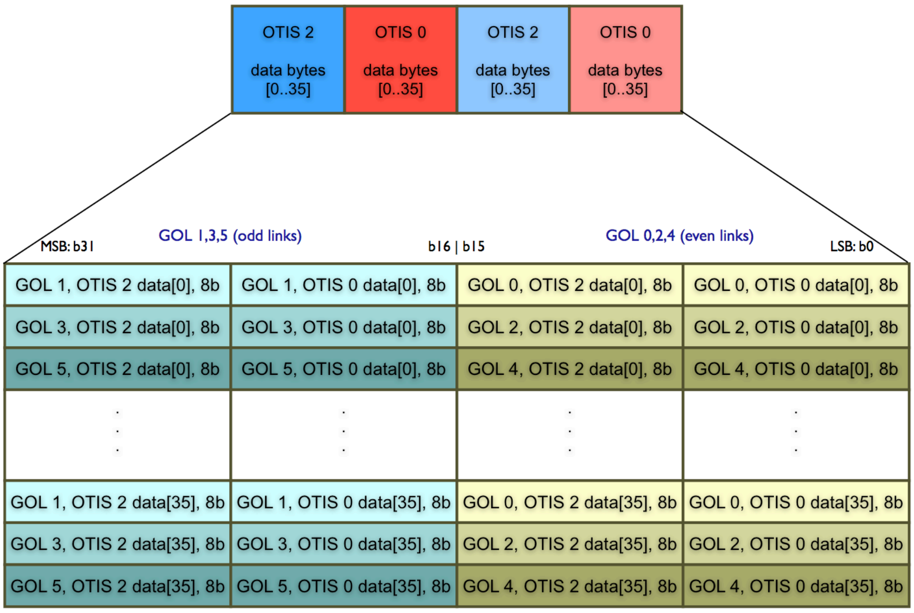
Figure 6: Outer Tracker RAW data bank structure

The amount of data in the "even" and "odd" data block is:

$$36\,\text{bytes/OTIS} \times 2\,\text{OTIS/link} \times 6\,\text{links} = 432\,\text{bytes} = 108\,\text{words}.$$

The actual sequence of the data is shown in Fig. 7 e.g. for the "even" OTIS' block: links are multiplexed in three consecutive 32-bit words, links 0, 2, and 4 using bits 15..0, and links 1, 3, and 5 using bits 31..16.

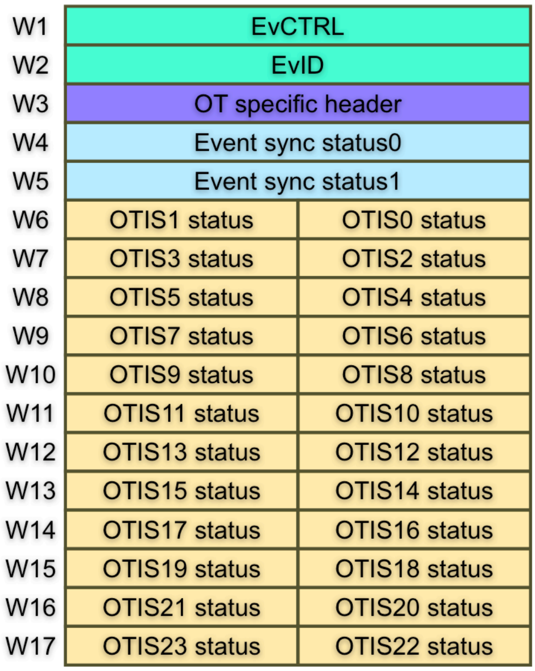
The total length of the RAW bank will depend on the number of enabled PP FPGAs. The TELL1 board has 4 PP FPGAs, but only 2 are connected to the Front-end Electronics of the Outer Tracker. Therefore, normally only two PP are enabled, while the data block of the two unused (disabled) PP FPGAs will not be sent (for debugging and commissioning all 4 PP FPGAs may be used).


OTIS 2 data bytes [0..35]
OTIS 0 data bytes [0..35]
OTIS 2 data bytes [0..35]
OTIS 0 data bytes [0..35]
MSB: b31
GOL 1,3,5 (odd links)
b16 | b15
GOL 0,2,4 (even links)
LSB: b0

GOL 1, OTIS 2 data[0], 8b	GOL 1, OTIS 0 data[0], 8b	GOL 0, OTIS 2 data[0], 8b	GOL 0, OTIS 0 data[0], 8b
GOL 3, OTIS 2 data[0], 8b	GOL 3, OTIS 0 data[0], 8b	GOL 2, OTIS 2 data[0], 8b	GOL 2, OTIS 0 data[0], 8b
GOL 5, OTIS 2 data[0], 8b	GOL 5, OTIS 0 data[0], 8b	GOL 4, OTIS 2 data[0], 8b	GOL 4, OTIS 0 data[0], 8b
⋮	⋮	⋮	⋮
GOL 1, OTIS 2 data[35], 8b	GOL 1, OTIS 0 data[35], 8b	GOL 0, OTIS 2 data[35], 8b	GOL 0, OTIS 0 data[35], 8b
GOL 3, OTIS 2 data[35], 8b	GOL 3, OTIS 0 data[35], 8b	GOL 2, OTIS 2 data[35], 8b	GOL 2, OTIS 0 data[35], 8b
GOL 5, OTIS 2 data[35], 8b	GOL 5, OTIS 0 data[35], 8b	GOL 4, OTIS 2 data[35], 8b	GOL 4, OTIS 0 data[35], 8b

Figure 7: Data arrangement in the RAW bank

The Event Information section is used to qualify and debug an event, mainly providing detailed information about the event fragments coming from the OTIS chip. Its format is given in Fig. 8 and Tab. 6.



W1	EvCTRL	
W2	EvID	
W3	OT specific header	
W4	Event sync status0	
W5	Event sync status1	
W6	OTIS1 status	OTIS0 status
W7	OTIS3 status	OTIS2 status
W8	OTIS5 status	OTIS4 status
W9	OTIS7 status	OTIS6 status
W10	OTIS9 status	OTIS8 status
W11	OTIS11 status	OTIS10 status
W12	OTIS13 status	OTIS12 status
W13	OTIS15 status	OTIS14 status
W14	OTIS17 status	OTIS16 status
W15	OTIS19 status	OTIS18 status
W16	OTIS21 status	OTIS20 status
W17	OTIS23 status	OTIS22 status

Figure 8: Format of the Event Information Section in the RAW Bank

W1 (EvCTRL)

Bit(31)	general error
Bit(30)	data generator enable (TELL1 internal data generator)
Bit(29)	ECS trigger (trigger not generated by TTC)
Bit(28..24)	not used, all 0
Bit(23..21)	trigger type
Bit(20..16)	bank list (’0’ & RAW bank & ”111”)
Bit(15..12)	detector ID (0x3 for OT)
Bit(11..0)	bunch counter

W2 (EvID)

Bit(31..0)	32 bit L0 event counter

W3 (OT specific header)

Bit(31..28)	not used, all 0
Bit(27..25)	trigger type
Bit(24)	general error flag for this PP FPGA
Bit(23..16)	bunch counter
Bit(15..0)	number of enabled GOLs in this PP FPGA

W4 (Event sync status0) (for the 6 bit wide vector, one bit for each link)

Bit(31..30)	PP FPGA chip address (”00” for PP0, ... ”11” for PP3)
Bit(29..24)	event buffer full (the rx-buffer is full for this event)
Bit(23..18)	event buffer empty (the rx-buffer is empty for this event)
Bit(17..12)	event size error (*still not working in version v1.8*)
Bit(11..6)	TLK transmission error
Bit(5..0)	GOL ID not equal

W5 (Event sync status1) (for the 6 bit wide vector, one bit for each link)

Bit(31..18)	not used, all 0
Bit(17..12)	GOL has hits
Bit(11..6)	optical link clock not active
Bit(5..0)	link (GOL) disabled

W6-W17 contains the OTIS chip status (16 bits for each OTIS)

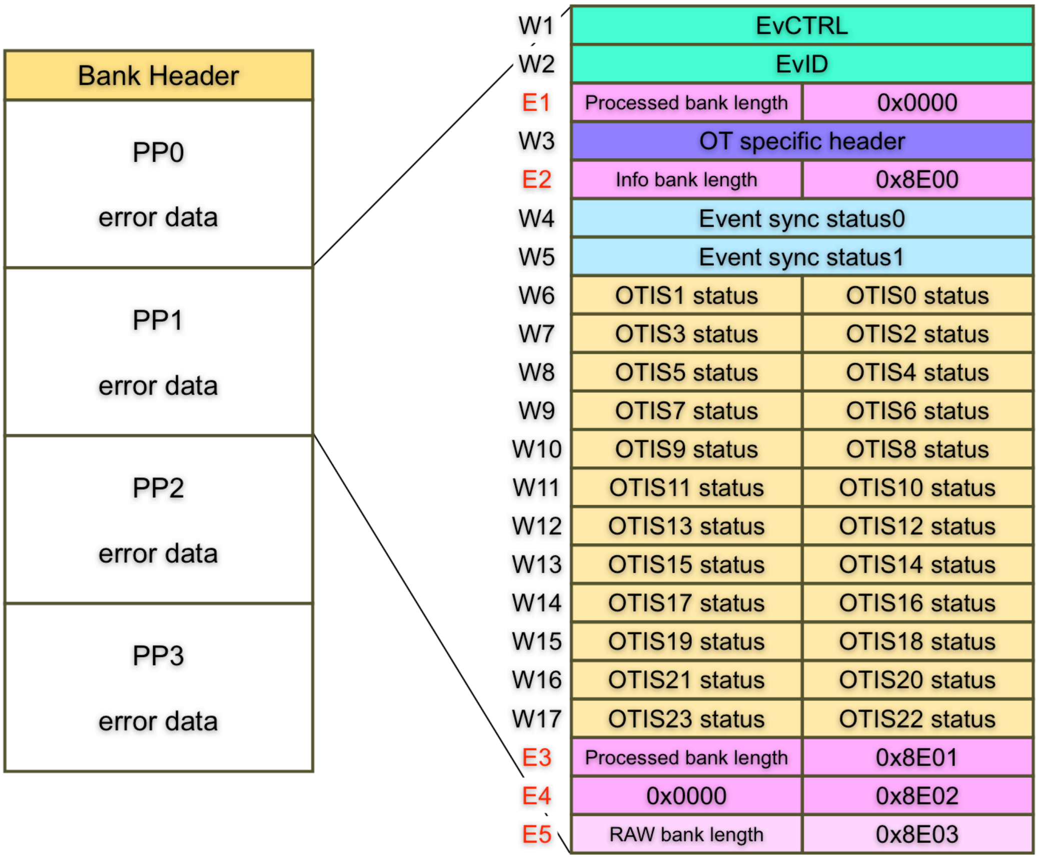
Bit(15..14)	not used, all 0
Bit(13)	set to 1, if bit(19) in the received OTIS header is not ’1’
Bit(12)	OTIS disabled
Bit(11)	OTIS bunch counter not equal to the one from the TTC
Bit(10)	OTIS event counter not equal to the one from the TTC
Bit(9)	OTIS ID wrong (not ”00”... ”11” for OTIS 0 ... 3)
Bit(8)	expected OTIS ID wrong
Bit(7)	OTIS offline
Bit(6)	OTIS offline zero (no ID different from 0x000 found)
Bit(5..0)	number of hits in this OTIS

Table 6: Event information block in detail

3.3 Error Bank

The format of the Error Bank is illustrated in Fig. 9. It starts with the bank header shown in Fig. 4 (Bank Type 0x21 identifies an Error Bank) followed by the errors from the **enabled** PP FPGAs.

The detailed meaning of the various “E”-fields is illustrated in Tab. 7. Notice that the content of the fields W1-W17 is identical to that in the Event Info block of the RAW bank (see Sect. 3.2).



W1	EvCTRL	
W2	EvID	
E1	Processed bank length	0x0000
W3	OT specific header	
E2	Info bank length	0x8E00
W4	Event sync status0	
W5	Event sync status1	
W6	OTIS1 status	OTIS0 status
W7	OTIS3 status	OTIS2 status
W8	OTIS5 status	OTIS4 status
W9	OTIS7 status	OTIS6 status
W10	OTIS9 status	OTIS8 status
W11	OTIS11 status	OTIS10 status
W12	OTIS13 status	OTIS12 status
W13	OTIS15 status	OTIS14 status
W14	OTIS17 status	OTIS16 status
W15	OTIS19 status	OTIS18 status
W16	OTIS21 status	OTIS20 status
W17	OTIS23 status	OTIS22 status
E3	Processed bank length	0x8E01
E4	0x0000	0x8E02
E5	RAW bank length	0x8E03

Figure 9: Error bank structure

E1

Bit(31..16)	length of the processed bank in this PP FPGA (in bytes)
Bit(15..0)	constant value 0x0000

E2

Bit(31..16)	If this PP FPGA is used and the event part is not correct, this field is 0x0038. Otherwise it is 0x0000 and the words W4-W17 will not exist.
Bit(15..0)	constant value 0x8E00

E3

Bit(31..16)	length of the processed bank in this PP FPGA (in bytes)
Bit(15..0)	constant value 0x8E01

E4

Bit(31..0)	constant value 0x00008E02

E5 (This word is added only if the RAW bank exists)

Bit(31..16)	RAW bank length 0x03A4 if this PP FPGA is used. Otherwise 0x0000.
Bit(15..0)	constant value 0x8E03

Table 7: Content of the “E” fields in the Error Bank

4 Event size calculation

The TELL1 board is provided with four 1 Gigabit Ethernet ports. At a maximum L0 trigger rate of 1.11111 MHz it could then support a data throughput of:

$$\frac{4 \times 10^9\,\mathrm{bits/s/event}}{1.11111 \times 10^6\,\mathrm{s}^{-1}} \simeq 3600\,\mathrm{bits/event} = 450\,\mathrm{bytes/event}.$$

However, considering the network payload, the process costs and adding a safety factor, one can assume a data throughput of 320 bytes (80 32-bit words) per event under full trigger rate.

The event size of an OT MEP can be estimated assuming that each TELL1 board will have 9 optical links connected.

Hitmap mode

As shown in Tab. 8, a TELL1 producing processed data in hitmap mode will provide a one-event MEP with a size of 197 bytes, assuming that the MEP factor is set to 12 and that no errors are generated. This is compatible with the TELL1 output bandwidth of 320 bytes per event.

	3 words (MEP header)/12 (MEP factor)
+	1 word (MEP sub-header)
+	2 words (bank header)
+	1 word (OT specific header)
+	9 words (9 GOL headers)
=	13.25 words
+	9x4 words (hitmap words for each GOL)
=	49.25 words
=	197 bytes/event

Table 8: Event size calculation in hitmap mode

Zero-suppress mode

In zero-suppress mode, while the total size of the various headers is the same as in hitmap mode, the total size of the drift-time data will depend on the detector occupancy:

$$\begin{aligned}
N_D \left[\frac{\text{words}}{\text{TELL1}}\right] &= 9\left[\frac{\text{Links}}{\text{TELL1}}\right] \times (\text{Occupancy}\,[\%] \times 128)\left[\frac{\text{Hits}}{\text{Link}}\right] \times 2\left[\frac{\text{bytes}}{\text{Hit}}\right] \\
&= (576 \times \text{Occupancy}\,[\%])\left[\frac{\text{words}}{\text{TELL1}}\right].
\end{aligned}$$

Table 9 summarizes the size of a one-event MEP in zero-suppress mode. A MEP size of 80 words, compatible with the TELL1 output bandwidth of 320 bytes per event, corresponds then to an average occupancy of **11.6%**.

	3 words (MEP header)/12 (MEP factor)
+	1 word (MEP sub-header)
+	2 words (bank header)
+	1 word (OT specific header)
+	9 words (9 GOL headers)
=	13.25 words
+	(576 × Occ [%]) words
=	13.25 + (576 × Occ [%]) words

Table 9: Event size calculation in zero-suppress mode

Error Banks

Since by default only two PP FPGAs are connected to the OT Front-end Electronics and enabled, the size of an Error Bank will be of 60 words, as shown in Tab. 10. Once an Error Bank is generated, its size adds up to the MEP size, the additional data throughput depending of course on the error rate.

+	2 words (bank header)
+	2 × 21 words (enabled PP FPGAs)
+	2 × 8 words (disabled PP FPGAs)
=	60 words
=	240 bytes/event

Table 10: Calculation of the data size added by each Error Bank

5 Compatibility with LHCb-2004-033 [2]

The OT DAQ data format described in this note is based on that defined in Ref. [2] (see Tab. 11), with a few changes introduced to satisfy the requirements of the LHCb DAQ data format [1, 4], which are summarized in Tab. 12.

Object	Content			
Standard Header	Bank Size [31..0] Magic Pattern [31..0] Source ID: Bank ID [31..16] Bank Type: OT [15..0]			
OT DATA	TELL1 HEADER	L1-ID[31..0]		
		R[31..24], L0-ID[23..0]		
		ErrorProgation from TELL1 [31..24] Trigger-Type[23..20] Bunch ID [19..8] Number of GOLs in the Bank [7..0]		
	TELL1 DATA	GOL 0	GOL Header	GOL 0 Header
			GOL 0 Data	Hit Data 0, Hit Data 1 Hit Data 2, Hit Data 3, ...
		GOL 1	GOL Header	GOL 1 Header
			GOL 1 Data	Hit Data 0, Hit Data 1 Hit Data 2, Hit Data 3, ...
		...		
		...		

Table 11: OT DAQ data format described in LHCb-2004-033 [2]

LHCb-2004-033	**Now**	**Comment**
Bank size (32b)	Bank length (16b) in Bank header	16 bits = maximum length of 64 kB
Magic pattern (32b)	Magic pattern 0xCBCB (16b) in bank header	see Ref. [1]
Source ID (16b)	Source ID (16b) in bank header	LHCb-wide definition
Bank type (16b)	Version + Type (16b) in bank header	LHCb-wide definition
L1-ID (32b)	Removed	No L1 trigger in the 1 MHz readout scheme
L0-ID (24b)	L0-EvID (16b) for each event in MEP sub-header, L0-EvID (32b) of the first event inside the MEP in MEP header.	
Error propagation (8b)	A general error flag bit in the OT specific header, detailed information in the error bank.	The error propagation is seperated into a flag and the error bank.
Trigger type (4b)	Trigger type (3b) in OT specific header	TTC uses only 3 bits for the trigger type
Bunch ID (12b)	Bunch counter (8b) in OT specific header	OTIS sends only 8 bits BxID
Number of GOL (8b)	Number of GOL (16b) in OT specific header	To be compatible with the current TELL1 framework
GOL header	GOL header	Hit number is reduced to 8b (0-255). One bit saved to identify the the data format of this GOL block
GOL data	GOL data	hitmap mode **OR** zero-suppress mode

Table 12: Changes from LHCb-2004-033 [2] to the present note

6 Debugging support

For debugging and commissioning the following features are provided:

- RAW bank: if requested by the "force_raw_bank" bit in the TELL1 configuration file or dynamically by trigger type 0x5, the unprocessed ("raw") data bank will be sent in addition to the processed bank. Using the raw data, one can re-process them offline to cross-check the TELL1 internal algorithm. Given the large event size (see Sect. 4), the trigger rate should not exceed 10 kHz.
- Error bank: The error bank has all the detailed information, e.g. event synchronisation result, OTIS status etc. In normal running conditions, this bank is only sent out if an error is detected, but it can be forced by setting a bit in a TELL1 control register.
- Histograms: Hit histograms (per channel) and drift time distributions (per OTIS) are filled in the TELL1 and made available via the Experiment Control System (ECS).

7 C program example

A C function to parse the MEP block and each data field can be found in the TELL1 CCPC support library, which can be downloaded from:

http://lphe.epfl.ch/ ghaefeli/VHDL Framework.htm

Download the c_code_only.zip of the latest release version available and find the "parse_mep" function in the "tell1lib/tell1offline.h".

There is also a C++ class called "OTevent" that provides a number of methods to read MEPs from a buffer or file, perform a number of cross checks and write out the data to file in various formats. It can be used both in the on- (e.g. gbe_sniffer) and off-line software. This class is not available in the tell1lib yet, but it can be requested from Mirco Nedos (mnedos@physik.uni-dortmund.de).

References

[1] O. Callot, M. Cattaneo, R. Jacobsson, B. Jost, P. Mato, N. Neufeld
"Raw-data Format", EDMS 565851

[2] J. Nardulli, A. Pellegrino, D. Wiedner
"Outer Tracker DAQ Data Format", LHCb-2004-033;
J. Nardulli, A. Pellegrino, D. Wiedner
"Outer Tracker Level 1 Data Format", LHCb-2005-031

[3] A. Pellegrino, U. Uwer, D. Wiedner
"Address Scheme for the Outer Tracker FE Electronics"
LHCb-2003-041

[4] B. Jost, N. Neufeld
"Raw-data transport format", EDMS 499933

[5] G. Haefeli, A. Gong
"VELO and ST error bank data format", EDMS 694818

[6] G. Haefeli, A. Gong
"VELO and ST non-zero-suppressed bank data format", EDMS 692431

[7] G. Haefeli, A. Gong
"VELO and ST zero-suppressed bank data format", EDMS 690583