RAW DATA FORMAT

version 4.0

Records in the data stream:

SOR	Start of run record.
EVENT	Event record.
RCU	Each event record consists of RCU data, each RCU has a header.
DATA	Event data of ALTRO format for each RCU.
EOR	End Of Run record.
BOF	If local data logging. If the file size exceeds a maximum size then the file is closed and a new file is opened and the first record in the new file is a Beginning Of File record.
EOF	If local logging. If the file size exceeds a maximum size then the file is closed and a new file is opened. The last record in the closed file is End Of File record.
POR	Pause run record
COR	Continue run record

If the data logging is done locally, then the filename is: readout-<runnb>_<filenb>.dat, where runnb is the run number, and <filenb> is a file counter, incremented with one for each file opened within the current run, and starting from 0 for the first file.

START OF RUN FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_SOR (=0x11111111)		
Data format version		
Run number		
Year	Month	Day
Hour	Minute	Second

END OF RUN FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_EOR (=0x33333333)		
Number of events		
Year	Month	Day
Hour	Minute	Second

PAUSE RUN FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_POR (=0x11112222)		
Number of events		
Year	Month	Day
Hour	Minute	Second

CONTINUE RUN FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_COR (=0x11113333)		
Number of events		
Year	Month	Day
Hour	Minute	Second

RAW EVENT FORMAT (32-bit words)
NOTE: DBOX DATA NOT YET IMPLEMENTED

Total event length (exclusive, added by software)
Header length (exclusive, added by software)
Block identifier = BLOCK_EVENT (=0x22222222) (added by software)
Software event number (incremented by software for each read event)
DBOXEVTN DBOX hardware trigger number (read from distributor box)
DBOXTIME DBOX time stamp (read from distributor box)
TLUEVTNM TLU event number (read from distributor box)
RCU block length (exclusive, added by software)
RCU identifier (added by software)
RCU HEADER – 8 words
ALTRO DATA – N40 = # of 40 bit words = (N40*5)/4 32 bit words = N32
...
...
RCU block length (exclusive, added by software)
RCU identifier (added by software)
RCU HEADER – 8 words
ALTRO HW DATA – N40 40 bit words = (N40*5)/4 32 bit words = N32
RCU TRAILER

RCU HEADER

<table>
<tr><td colspan="6">BLOCK LENGTH [31..0] = FFFFFFFF</td></tr>
<tr><td colspan="2">FORMAT [31..24] = 1</td><td colspan="2">L1 Type [21..14]</td><td>[13:12] = 0</td><td>EVT ID1 [11..0] = 0</td></tr>
<tr><td colspan="2">[31..24] = 0</td><td colspan="4">EVT ID2 [23..0] = N => 0 ??</td></tr>
<tr><td colspan="2">Block [31..24] = 0</td><td colspan="4">Participating sub-detectors [23..0] = 0</td></tr>
<tr><td>[31..28] = 0</td><td colspan="4">Status/Error [27..12]</td><td>Bunch [11..0]</td></tr>
<tr><td colspan="6">Trigger classes low [31..0] = 0</td></tr>
<tr><td>ROI [31..28]</td><td colspan="2">[27..18] = 0</td><td colspan="3">Trigger classes high [17..0] = 0</td></tr>
<tr><td colspan="6">Region Of Interest (ROI) [31..0]</td></tr>
</table>

ALTRO HW 40 bit word DATA example for one channel:

40 30 20 10

S05	S04	S03	S02 (sample)
S10	007 (length)	T06 (time stamp)	S06
005	T12	S12	S11

....

S91	S90	S89	S88 (sample)
2AA	007 (length)	T92 (time stamp)	S92

2AAA (14 -bits)	# 10 bit words (10 bits)	A (4 bits)	12-bit hardware address

Code [31:26]	Name	Description [25:0]
0x00	PAYLOAD LENGTH	The payload length is expressed {No. of 40 bit words for DM} {No. of 32 bit words for RM}
0x01	ERR_REG1	Trailer[25:13] ≡ FECERRA[19 :7]; Trailer[12:0] ≡ FECERRB[19 :7]; FECERR[19] : *transfer* not released FECERR[18]: ALTRO *error* signal asserted while data being transferred; FECERR[17]:*transfer* not asserted; FECERR[16]: ALTRO *error asserted* before *transfer*; FECERR[15]: Write FSM error when started by RDO FSM; FECERR[14]: *ackn* not released by ALTRO; FECERR[13]: ALTRO *error* asserted while waiting for *ackn* to be released; FECERR[12]: ALTRO does not assert *ackn*; FECERR[11]: ALTRO *error* asserted while waiting for *ackn*; FECERR[10]: ALTRO asserts *ackn* before assertion of *cstb*; FECERR[09]: ALTRO *error* asserted while waiting for *ackn* in state "chkack"; FECERR[08]: ALTRO error asserted in state "assertbus" FECERR[07]: ALTRO error asserted in the execution of a broadcast command
0x02	ERR_REG2	ERR_REG2[8] : Block Length mismatch ERR_REG2[7]: Channel Address mismatch; ERR_REG2[6]: RDYRX error; ERR_REG2[5]: SCANEVLEN error; ERR_REG2[4:0]:EVLENRDO error;
0x03	ERR_REG3	ERR_REG3 [11:0] ≡ Number of mismatches in the channels address detected by the data assembler during the readout ERR_REG3[24:12] = Number of mismatches in the channel data block length detected by the data assembler during the readout.
0x04	ERR_REG4	NOT USED "00000000000….0000000000000"
0x05	FEC_RO_A	It defines the bit map of the "active" FECs of Branch A.
0x6	FEC_RO_B	It defines the bit map of the "active" FECs of Branch B.
0x7	RDO_CFG1	Copy of the RCU Register ALTROCFG1. For the definition of this parameter, see also ALTRO manual pags. 36,37. First Baseline Correction Mode ≡ RDO_CFG1[3:0] Polarity. When set, the ADC data is inverted (1's C) [4] Nr. Of pre-samples excluded from 2nd baseline corr. ≡ RDO_CFG1 [6:5] Nr. Of post-samples excluded from 2nd baseline corr. ≡ RDO_CFG1 [10:7] Enable second baseline correction ≡ RDO_CFG1 [11] Glitch filter configuration for zero suppression ≡ RDO_CFG1 [13:12] Nr. Of post-samples excluded from suppression ≡ RDO_CFG1[16:14] Nr. Of pre-samples excluded from suppression ≡ RDO_CFG1[18:17] Enable Zero Suppression ≡ RDO_CFG1 [19]
0x8	RDO_CFG2	Nr of ALTRO Buffers (copy of RCU register ALTROCFG2[4]) ≡ RDO_CFG2[24] Nr of pre-trigger samples (copy of RCU reg ALTROCFG2[[3:0]) ≡ RDO_CFG2[23:20] Nr. Samples / channel (copy of RCU reg ALTROIF[9:0]) ≡ RDO_CFG2[19:10] Sparse Readout ≡ RDO_CFG2[9] $T_{sampling}$ / T_{LHC} ≡ RDO_CFG2[8:5] ➔ 00 = 2 (20MHz), 01 = 4 (10MHz), 10 = 8 (5 MHz) Phase of L1 trigger wrt LHC bunch crossing ≡ RDO_CFG2 [4:0]. The phase is internally calculated in terms of bunch crossing cycles. For example if the Tsampling/TLHC is equal 01 (10MHz sampling rate), only the two least significant bits are meaningful and the phase can take values 0, 1, 2, 3.
0x9	RCU ID	[25:16] ≡ 0x2AA ; [15:7] ≡ RCU Address Add[8] = 0 for A-side, 1 for C-side Add[7:3] = sector number (0:17) Add[2:0] = readout partition number (0:5) [6:0] ≡ trailer length (nr. of 32-bit words)

RCU TRAILER DATA FORMAT

End of data structures depending on the number of 40 bit ALTRO data words. There is a description of the format later in this document.

N32 modulus 5 = 0

<table>
<tr><td colspan="4">ALTRO WORD1 [31..0]</td></tr>
<tr><td colspan="3">ALTRO WORD2 [23..0]</td><td>ALTRO WORD1 [39..32]</td></tr>
<tr><td colspan="2">ALTRO WORD3 [15..0]</td><td colspan="2">ALTRO WORD2 [39..24]</td></tr>
<tr><td>ALTRO WORD4 [7..0]</td><td colspan="3">ALTRO WORD3 [39..16]</td></tr>
<tr><td colspan="4">ALTRO WORD4 [39..8]</td></tr>
</table>

N32 modulus 5 = 2

<table>
<tr><td colspan="4">ALTRO WORD1 [31..0]</td></tr>
<tr><td colspan="3">AAAAAAA</td><td>ALTRO WORD1 [39..32]</td></tr>
</table>

N32 modulus 5 = 3

<table>
<tr><td colspan="4">ALTRO WORD1 [31..0]</td></tr>
<tr><td colspan="3">ALTRO WORD2 [23..0]</td><td>ALTRO WORD1 [39..32]</td></tr>
<tr><td colspan="2">AAAA</td><td colspan="2">ALTRO WORD2 [39..24]</td></tr>
</table>

N32 modulus 5 = 4

<table>
<tr><td colspan="4">ALTRO WORD1 [31..0]</td></tr>
<tr><td colspan="3">ALTRO WORD2 [23..0]</td><td>ALTRO WORD1 [39..32]</td></tr>
<tr><td colspan="2">ALTRO WORD3 [15..0]</td><td colspan="2">ALTRO WORD2 [39..24]</td></tr>
<tr><td>AA</td><td colspan="3">ALTRO WORD3 [39..16]</td></tr>
</table>

BEGINNING OF FILE FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_BOF (=0x44444444)		
Data format version		
Run number		
Year	Month	Day
Hour	Minute	Second

File number
Last event number

END OF FILE FORMAT

Total length (exclusive)		
Header length (exclusive)		
Block identifier = BLOCK_EOF (=0x55555555)		
Last event number		
Year	Month	Day
Hour	Minute	Second

Last file number

RCU data format over the DDL

The RCU data formatter converts the blocs of 4x40 bit altro words into blocks of 5 32 bit words. One such block will from now on be called a DDL block. In general the number of 40 bit altro words of a RCU payload is not divisible by 4, therefore the last DDL data block might have either 2,3 or 4 32 bit words. It is necessary to find the position of the last altro word since the RCU payload is a back linked list of altro data. The position of the last 40 bit altro word can be found by taking the size of the payload modulus 5. more specific (sizeof(ddlbuffer)/4 − header size − Ntrailerwords)mod5. The mod 5 value of the DDL buffer determines unambiguous the position of the last altro word and thereby the total number of altro word. In addition the RCU trailer contains the count of 40 bit altro words that can e used as a consistency check. Given the payload size mod 5 value the position of the last altro word can be determined as given below.

• **Segmentation 0:** The number of 40 bit altro words is divisible by 4, the last DDL block is filled according to the table below. Position of last altro word when the a DDL block is exactly filled i.e the number of altro words is divisible by 4. This corresponds to the case when N40bitwords mod 4 =0, or alternatively N32altroPayloadsize mod 5 = 0

N32 modulus 5 = 0

ALTRO WORD1 [31..0]			
ALTRO WORD2 [23..0]			ALTRO WORD1 [39..32]
ALTRO WORD3 [15..0]		ALTRO WORD2 [39..24]	
ALTRO WORD4 [7..0]	ALTRO WORD3 [39..16]		
ALTRO WORD4 [39..8]			

• **Segmentation 1;** The number of 40 bit altro words modulus 4 equals one, the DDL block is filled according to the table below. Position of the last altroword of the last DDL block in the case where the last DDL block is filled with just one 40 bit word. In this case the last DDL block consists of two 32 bit words. The least significant 24 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAAAAAA. This corresponds to the case when N40bitwords mod 4 = 1, or alternatively N32altroPayloadsize mod 5 = 2

N32 modulus 5 = 2

ALTRO WORD1 [31..0]	
AAAAAAA	ALTRO WORD1 [39..32]

• **Segmentation 2;** The number of 40 bit altro words modulus 4 equals two, the DDL block is filled according to the table below. Position of the last altro word of the last DDL block when the last DDL block is filled with two 40 bit words. In this case the the last DDL block consists of three 32 bit words. The least significant 16 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAAAA. This corresponds to the case when N40bitwords mod 4 = 2, or alternatively N32altroPayloadsize mod 5 = 3

N32 modulus 5 = 3

ALTRO WORD1 [31..0]		
ALTRO WORD2 [23..0]		ALTRO WORD1 [39..32]
AAAA	ALTRO WORD2 [39..24]	

• Segmentation 3; The number of 40 bit altro words modulus 4 equals three, the DDL block is filled according to the table below. Position of the last altro word of the last DDL block when the last DDL block is filled with three 40 bit words. In this case the the last DDL block consists of four 32 bit words. The least significant 8 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAA. This corresponds to the case when N40bitwords mod 4 = 3, or alternatively N32altroPayloadsize mod 5 = 4

N32 modulus 5 = 4

<table>
<tr><td colspan="4">ALTRO WORD1 [31..0]</td></tr>
<tr><td colspan="3">ALTRO WORD2 [23..0]</td><td>ALTRO WORD1 [39..32]</td></tr>
<tr><td colspan="2">ALTRO WORD3 [15..0]</td><td colspan="2">ALTRO WORD2 [39..24]</td></tr>
<tr><td>AA</td><td colspan="3">ALTRO WORD3 [39..16]</td></tr>
</table>