FlexRay

ASC Logging Format

Specification

Version 1.4.6 of 2017-08-23

Status	Completed
Publisher	Vector Informatik GmbH

Document Management

Revision list

Version	Date	Editor	Section	Changes, comments
1.3	2005-05-09	Hb	-	Header byte corrected
1.4	2005-08-30	Wbn	-	Format description for new events
1.5	2006-04-06	Wbn	-	Event flags and descriptions updated
1.6	2006-05-11	Wbn	-	Improvements
1.7	2006-10-25	Wbn	-	Some minor updates in descriptions, example updated
1.8	2006-11-10	Hb	-	Some minor changes
1.9	2007-01-16	Wbn	-	Some minor changes
1.10	2007-03-19	Jsd	-	Some minor changes
1.11	2007-05-24	Wbn	-	Examples updated
1.20	2007-10-18	Wbn	-	PDUs added
1.21	2007-12-17	Hb	-	Small fix in PDU-description
1.3	2008-09-16	Wbn	3	Update
1.31	2008-12-04	Wbn	3.3.5	Bit 22 added, description for bit 17 updated
1.4.1	2009-05-19	Wbn	1	Added Disclaimer
1.4.2	2009-09-08	Wbn	4.3.4	Tx Conflict (Bit 12) for VN interfaces added for CANoe/CANalyzer 7.2
1.4.3	2015-04-27	Hb	4.3.4	Additional error flags (since CANoe/CANalyzer 8.5 SP3)
1.4.4	2015-06-23	Vrd	3.4	Modified channel range
1.4.5	2016-12-19	Mom	all	CI and layout
1.4.6	2017-08-23	Yav	4.7.1	FlexRay Status Event, POC State

Contents

1 Disclaimer

Severability clause - Restrictions for the usage of Vector logging data formats outside of Vector products

The format specification / access functions for the Vector BLF and ASC logging data formats are made available under the restrictions and conditions cited hereafter.

Please note that Vector Informatik neither gives any guarantee nor assumes any liability beyond compulsory legal regulations for the BLF or ASC logging format respectively as well as for the access functions to the single objects.

Vector Informatik disclaims all liability for errors which might be contained in the access functions or the format specification itself.

Vector Informatik does neither provide support for the integration into your software nor for problems occurring inside your software on the customer side.

Beyond that Vector Informatik reserves the right to change the BLF or ASC data format respectively anytime without prior notification. Therefore, the compatibility of the format is not ensured.

2 Overview

The document specifies the format of FlexRay events in the CANoe/CANalyzer ASC logging.

3 Old Format

The “Old Format” is applied for FlexRay events, which are implemented in CANoe/CANalyzer version 5.1 and below.

3.1 Header

Events can either be recorded in hexadecimal (“base hex”) or decimal mode (“base dec”). Timestamps are written absolute (“timestamps absolute”) or relative to the preceding event (“timestamps relative”). See section 3.2 for an example.

3.2 Example

The following logging contains FlexRay Message events and FlexRay StartCycle Events.

```
date Fri Feb 25 12:25:32 am 2005
base dec  timestamps absolute
internal events logged
Begin Triggerblock Fri Feb 25 12:25:32 am 2005
   0.0000 Start of measurement
   0.0367 Fr * StartCycleEvent NM Vector: 2 00 00
   0.0368 Fr 2 V9  2 24 0 1 1193 x  4   2 238   2  87 01d0  9a
   0.0370 Fr 2 V9  4 24 0 1 151 x  4  20 238  22  22 01d0  9a
   0.0376 Fr 2 V9  13 24 0 0 620 x  4   2  88   0  13 0180  88
   0.0377 Fr 2 V9  14 24 0 0 115 x  4   2  88   0  14 0180  88
   0.0377 Fr 2 V9  15 24 0 0 390 x  4   2  88   0  15 0180  88
   0.0378 Fr 2 V9  16 24 0 0 959 x  4   2  88   0  16 0180  88
   0.0379 Fr 2 V9  17 24 0 0 586 x  4   2  88   0  17 0180  88
   0.0379 Fr 2 V9  18 24 0 0 85 x  4   2  88   0  18 0180  88
   0.0380 Fr 2 V9  19 24 0 0 416 x  4   2  88   0  19 0180  88
   0.0381 Fr 2 V9  20 24 0 0 1131 x  4   2  88   0  20 0180  88
   0.0381 Fr 2 V9  21 24 0 0 1438 x  4   2  88   0  21 0180  88
   0.0382 Fr 2 V9  22 24 0 0 1921 x  4   2  88   0  22 0180  88
   0.0383 Fr 2 V9  23 24 0 0 1652 x  4   2  88   0  23 0180  88
   0.0393 Fr 2 V9  37 24 0 0 511 x  4   2  24   0  37 0180  88
   0.0393 Fr 2 V9  38 24 0 0 992 x  4   2  24   0  38 0180  88
   0.0394 Fr 2 V9  39 24 0 0 533 x  4   2  24   0  39 0180  88
   0.0395 Fr 2 V9  40 24 0 0 1011 x  4   2  24   0  40 0180  88
   0.0395 Fr 2 V9  41 24 0 0 518 x  4   2  24   0  41 0180  88
   0.0396 Fr 2 V9  42 24 0 0 25 x  4   2  24   0  42 0180  88
   0.0397 Fr 2 V9  43 24 0 0 492 x  4   2  24   0  43 0180  88
   0.0398 Fr 2 V9  44 24 0 0 1063 x  4   2  24   0  44 0180  88
   0.0398 Fr 2 V9  45 24 0 0 1490 x  4   2  24   0  45 0180  88
   0.0399 Fr 2 V9  46 24 0 0 1997 x  4   2  24   0  46 0180  88
   0.0400 Fr 2 V9  47 24 0 0 1592 x  4   2  24   0  47 0180  88
   0.0400 Fr 2 V9  48 24 0 0 1025 x  4   2  24   0  48 0180  88
   0.0417 Fr * StartCycleEvent NM Vector: 2 00 00
   0.0418 Fr 2 V9  2 25 0 1 1193 x  4   2 188   2  88 01d0  9a
   0.0418 Fr 1 V9  2 25 0 1 1193 Ident_02_Rnd_0  4   2 188   2  88 00d0  9a
   0.0420 Fr 2 V9  4 25 0 1 151 x  4  21  87  22 148 01d0  9a
   0.0420 Fr 1 V9  4 25 0 1 151 Ident_04_Rnd_0  4  21  87  22 148 00d0  9a
   0.0426 Fr 2 V9  13 25 0 0 620 x  4   2  89   0  13 0180  88
   0.0426 Fr 1 V9  13 25 0 0 620 x  4   2  89   0  13 0080  88
   0.0427 Fr 2 V9  14 25 0 0 115 x  4   2  89   0  14 0180  88End
TriggerBlock
```

3.3 FlexRay Events

The section lists all FlexRay events in CANoe/CANalyzer ASC logging. See section 3.4 for an explanation of the symbols.

FlexRay Message Event	
FlexRay Message received or transmitted on a FlexRay channel.	
Format	<Time> <Channel> <Typ> <Id> <Cycle> <NM> <Sync> <HeaderCRC> <SymbolicName> <DLC> <D0> <D1>...<D64> x<FrameState> x<HeaderBitMask>
Example	0.0420 Fr 1 V9 4 25 0 1 151 Ident_04_Rnd_0 4 21 87 22 148 00d0 9a 0.0426 Fr 2 V9 13 25 0 0 620 x 4 2 89 0 13 0180 88

FlexRay Start Cycle Event	
FlexRay StartCycle Event transmitted by the FlexCard	
Format	<Time> <Channel> <Typ> NM Vector: <DLC> <D0> <D_{DLC}>
Example	0.0417 Fr * StartCycleEvent NM Vector: 2 00 00

3.4 Symbols

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example	Special
<Time>		>=9	Absolute or relative time in seconds		1234.5678	usually 4 decimal places
<Channel>	6	6	The number of the FlexRay channel.	1-255,*	Fr 1, Fr 2, Fr *	'*' describes the channel wildcard
<Typ>	>=2	>=2	Type of this event	V9, StartCycle Event, ErrorFram e	V9	
<ID>	3	4	Numeric identifier	1…2047	50	
<Cycle>		2	Current number of communication cycle	0…63	10	
<NM>	1	1	Obsolete element not used in the V9	0..1	0	
<Sync>	1	1	Sync frame Indicator	0..1	1	
<HeaderCRC>	3	4	HeaderCRC	0…4095	1025	

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example	Special
<Symbolic-Name>	/	/	Database name of the frame	/	Ident_04_Rnd	'x' if not available
<DLC >	2	3	Payload length	0…254	5	
<Dx>	3	2	data byte x	0…255	0x1E	
<FrameState>	4	5	See table below		0180	
<HeaderBit-Mask>	2	3	FlexRay frame header: Bit Meaning 0 Payload Preample Ind. 1 Sync. Frame Indicator 2 Reserved Bit 3 Null Frame Indicator 4 Startup Frame Ind. 5-7 Frame State Format Mask (see below) Bit 5-7 Meaning (Hdr.Byte) 0 (0x00) Motorola V.6 1 (0x20) reserved 2 (0x40) BusDoctor 3 (0x60) reserved 4 (0x80) FlexCard Cy. 5 (0xa0) reserved 6 (0xc0) reserved 7 (0xe0) reserved		88	

Bit	FlexCard Cyclone	BusDoctor
0	TXCON - TX CONflict (TXCON)	Decoding Error (NRZ) (CODERR)
1	BVIOL - Boundary VIOLation (MASB)	TSS Violation Error (TSSVIOL)
2	CERR - Content ERRor (CERR)	Header CRC Error (HCRCERR)
3	SERR - Syntax ERRor (SERR)	Frame CRC Error (FCRCERR)
4	SUPF - StartUP Frame indication (SUPF)	Frame End Sequence Error (FESERR)
5	NULLF - NULL Frame indication (NF)	Symbol (SYMB)
6	SYNCF - SYNC Frame indication (SF)	Valid Frame (VAL)
7	VCE - Valid Communication Element (VAL)	Boundary Violation Error (MASB)
8	—	NIT Violation Error (NITVIOL)
9	—	Symbol Window Violation Error (SWVIOL)
10	—	Slot Overbooked Error (SOVERR)
11	—	Null Frame Error (INFE)
12	—	Syncframe or Start-up Error (ISFE)
13	—	Frame ID Error (FIDE)
14	—	Cycle Counter Error (CCE)
15	—	Static Payload Length Error (PLSE)

4 New Format

The “New Format” is applied for FlexRay events, which are implemented in CANoe/CANalyzer since version 5.2.26. The new Format uses a code string to be able to distinguish the events of each other.

4.1 Terms and Acronyms

Term	Definition
CC	Communication controller
NM	Network management

4.2 General

Note: unused members or flags are not used yet and must be always set to 0 if logging object is written by another application.

Each record contains a common header data and an event specific data, which is described below.

HEADER	Specific event data
0.020000 Fr SE 0 0 1 3	51 1 1 0 0 0 0

Logging files are parsed using methods of the stream I/O (Class: istream). Fixed size for column widths is not defined, but parameters are separated by one or several blanks. The number of blanks does not affect on the read-operation and the leading zeros are ignored.

The number format can be changed using the std::dec respectively std::hex constants, the stream operations are case-insensitive. Each line in the logging file describes one event.

4.3 Common Data Types

4.3.1 Direction Flags

- Rx
- Tx
- Tx Request

4.3.2 Channel Mask

0 = Reserved or invalid

1 = FlexRay Channel A

2 = FlexRay Channel B

3 = FlexRay Channels A and B

4.3.3 CC-Types

Communication controllers (CC-Types):

- 0 = Architecture independent
- 1 = Invalid CC type (for internal use only)
- 2 = Cyclone I
- 3 = BUSDOCTOR
- 4 = Cyclone II
- 5 = Vector VN interface
- 6 = VN-Sync-Pulse (only in Status Event, for debugging purposes only)

4.3.4 Controller Specific Frame State Information

Note: unused bits in frame status field are not used yet and must be always set to 0 if logging object is written by another application.

Bit	Cyclone I	BUSDOCTOR	Cyclone II	VN
0	TX Conflict (TXCON)	Decoding Error (CODERR)	Syntax Error (SERR)	Syntax Error (SERR)
1	Boundary Violation (BVIOL)	Violation Error (TSSVIOL)	Content Error (CERR)	Content Error (CERR)
2	Content Error (CERR)	Header CRC Error (HCRCERR)	Slot BoundaryViolation (BVIOL)	Slot BoundaryViolation (BVIOL)
3	Syntax Error (SERR)	Frame CRC Error (FCRCERR)	Empty Slot (SLEMPTY)	Empty Slot (SLEMPTY)
4	StartUP Frame indication (SUPF)	Frame End Sequence Error (FESERR)	Message Lost (MLOST)	Message Lost (MLOST)
5	NULL Frame indication (NF)	Symbol (SYMB)	Valid Frame (VAL)	Valid Frame (VAL)
6	SYNC Frame indication (SF)	Valid Frame (VAL)		TX Conflict (TXCON)
7	Valid Communication Element (VCE)	Boundary Violation Error (MASB)		Framing Error (FrmERR)
8		NIT Violation Error (NITVIOL)		Header CRC Error (HdrERR)
9		Symbol Window Violation Error (SWVIOL)		Frame CRC Error (FrmCRC)
10		Slot Overbooked Error (SOVERR)		Reserved Bit Error
11		Null Frame Error (INFE)		Tx Conflict (bus signal level failure during transmission)
12		Syncframe or Start-up Error (ISFE)		Redundancy Error (dual channel frame with different payload or header flags detected)
13		Frame ID Error (FIDE)		Bus Error (Spikes detected)
14		Cycle Counter Error (CCE)		Unknown error
15		Static Payload Length Error (PLSE)		

4.3.5 Frame Flags

Description of frame flags.

Bit	Description
0	1 = Null frame.
1	1 = Data segment contains valid data
2	1 = Sync bit
3	1 = Startup flag
4	1 = Payload preamble bit
5	1 = Reserved bit
6	1 = Error flag (error frame or invalid frame)
7	Reserved
8	Internally used in CANoe/CANalyzer.
9	Internally used in CANoe/CANalyzer.
10	Internally used in CANoe/CANalyzer.
11	Internally used in CANoe/CANalyzer.
12	Internally used in CANoe/CANalyzer.
13	Internally used in CANoe/CANalyzer.
14	Internally used in CANoe/CANalyzer.
15	1 = Async. monitoring has generated this event
16	1 = Event is a PDU
17	Valid for PDUs only. The bit is set if the PDU is valid (either if the PDU has no update bit, or the update bit for the PDU was set in the received frame).
18	Reserved
19	1 = Raw frame (only valid if PDUs are used in the configuration). A raw frame may contain PDUs in its payload
20	1 = Dynamic segment 0 = Static segment
21	This flag is only valid for frames and not for PDUs. 1 = The PDUs in the payload of this frame are logged in separate logging entries. 0 = The PDUs in the payload of this frame must be extracted out of this frame. The logging file does not contain separate PDU-entries.
22	Valid for PDUs only. The bit is set if the PDU has an update bit.

The reserved bits and the bits which are for internally CANoe/CANalyzer usage must be ignored from other applications. Other applications must set these bits to 0 when writing logging files.

4.4 Header

The timestamp records are the same like in older events. New events use new header format, which is described in the following table:

Element	Type	Format	Remarks
Timestamp	Float	dec	
Bus type	String	-	“FR” or “Fr”
Event type	String	-	“RMSG”, ...
Clusternr.	UInt32	dec	Number of cluster: channel number – 1. Conversion from CANoe while reading: if “0”, channel number is set to “1”, else Clusternr. = Channelnr.
Client-ID	UInt32	dec	Client index of send node. Must be set to 0 if file is written from other applications.
Channelnr.	UInt32	dec	Application channel
Channel mask	UInt32	dec	See 4.3.2

4.5 FlexRay Message Event (“RMSG / PDU”)

Description: FlexRay message or PDU received or transmitted on FlexRay bus.

Format:

Element	Type	Format	Remarks
Slot ID	UInt32	dec/hex	Slot ID of the frame. For PDUs only: This is the PDU-identifier for internal usage.
Cycle no.	UInt32	dec/hex	Cycle number
Direction	String		See 4.3.1
App. param.	UInt32	dec	Not used, reserved
Flags	UInt32	dec/hex	See 0
CC-Type.	UInt32	dec/hex	Type of communication controller, see 4.3.3
CC-Data	UInt32	dec/hex	Controller specific frame state information, see 4.3.4
Header CRC	UInt32	dec/hex	Header CRC
Name	String		Name of the Frame or PDU as defined in the database, “x” if frame is not defined in the database, or “FlexRay_Frame_[x|x|x]” for raw frames (owner frames in PDU configurations).
Payload length	Int32	dec/hex	Payload length in bytes
Buffer length	Int32	dec/hex	Number of bytes of the payload stored in Data[n]. If the CC-frame buffer was too small to receive the complete payload, then buffer length is smaller than payload length.
Data[n]	UInt32	dec/hex	n = Buffer length
Frame CRC	UInt32	dec/hex	Frame CRC (since CANoe 6.1)
Spy Flag	UInt32	dec	If 1, async. monitoring has generated this event (since CANoe 6.1)
Frame Length NS	UInt32	dec	Length of frame in ns (only valid for frames received in asynchronous mode, bit 15 is set in the frame flags, see also 0). Since CANoe 6.1
PDU Offset	Int32	Dec	**For PDUs only:** offset of PDU in an owner (raw) frame

4.5.1 Frame Example

HEADER	Specific event data
0.039255 Fr RMSG 0 0 1 1	4 7 Rx 0 14 5 32 151 Message_2 4 4 27 24 29 241 0 0 0

4.5.2 PDU Example

PDU and raw frame example (for configurations where PDUs are used):

HEADER	Specific event data
0.047906 Fr PDU 0 2 1 1	3 7 Tx 0 2313218 5 32 559 PDU_DEMO_1 1 1 0 0 0 0 40
0.047906 Fr RMSG 0 2 1 1	3 7 Tx 0 2640898 5 32 559 FlexRay_Frame_[3|0|1] 42 42 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 128 0 128 0 128 0 0 0 0 0 128 0 128 0 128 0 0 0 56 0 0 0

4.6 FlexRay Start Cycle Event ("SCE")

Description: FlexRay StartCycle event transmitted by the FlexRay hardware.

4.6.1 Controller Specific Information

Field	Cyclone I	Cyclone II	VN-Interface
Data[0]	Rate correction of CC, read from RCVR register	Sync correction of CC, read from RCV register	Sync correction of CC, read from RCV register
Data[1]	Offset correction of CC, read from OCVR register	Offset correction of CC, read from OCV register	Offset correction of CC, read from OCV register
Data[2]		Cycles with no correction, read from CCEV register	Cycles with no correction, read from CCEV register
Data[3]		Cycles with correction in passive mode, read from CCEV register	Cycles with correction in passive mode, read from CCEV register
Data[4]		Sync Frame status, read from SFS register	Sync Frame status, read from SFS register

4.6.2 Format Description

Element	Type	Format	Remarks
Cycle no.	UInt32	dec/hex	Cycle number
Direction	String		See 4.3.1
CC-Type.	UInt32	dec/hex	Type of communication controller, see 4.3.3
CC-Data[5]	UInt32	dec/hex	Driver flags for internal usage, see 4.6.1
Label	String	-	"NM_Vector", not used
NM_Vect_L	UInt32	dec/hex	Length of NM-Vector in bytes
NM_Vect[n]	UInt32	dec/hex	Array of data bytes (NM vector max. length)

4.6.3 Example

HEADER	Specific event data
0.044115 Fr SCE 0 0 1 3	8 Rx 5 0 4 0 0 0 NM_Vector: 0

4.7 FlexRay Status Event (“SE”)

Description: FlexRay Status event transmitted by the FlexRay hardware on changed bus state (e.g. if bus lost synchronization)

4.7.1 Controller Specific Information

CC-Type: Cylone I

Field	Description
Data[0]	Content of Protocol state register (PSR)
Data[1]	Content of Module config register (MCR0)

CC-Type: BUSDOCTOR

Field	Description
LOW-WORD of Data[0]	Symbol length
HI-WORD of Data[0]	Flags: 1 = possible CAS
Data[1]	Reserved

CC-Type: VN-Interface

Field	Description
Data[0]	POC State in the operation control phase: **Mask** / **Description** 0x00 FR_STATUS_DEFAULT_CONFIG 0x01 FR_STATUS_READY 0x02 FR_STATUS_NORMAL_ACTIVE 0x03 FR_STATUS_NORMAL_PASSIVE 0x04 FR_STATUS_HALT 0x05 FR_STATUS_MONITOR_MODE 0x0F FR_STATUS_CONFIG POC State in the wake-up phase: **Mask** / **Description** 0x10 FR_STATUS_WAKEUP_STANDBY 0x11 FR_STATUS_WAKEUP_LISTEN 0x12 FR_STATUS_WAKEUP_SEND 0x13 FR_STATUS_WAKEUP_DETECT POC State in the start-up phase: **Mask** / **Description** 0x20 FR_STATUS_STARTUP_PREPARE 0x21 FR_STATUS_COLDSTART_LISTEN 0x22 FR_STATUS_COLDSTART_COLLISION_RESOLUTION 0x23 FR_STATUS_COLDSTART_CONSISTENCY_CHECK 0x24 FR_STATUS_COLDSTART_GAP 0x25 FR_STATUS_COLDSTART_JOIN

Field	Description
	0x26 FR_STATUS_INTEGRATION_COLDSTART_CHECK 0x27 FR_STATUS_INTEGRATION_LISTEN 0x28 FR_STATUS_INTEGRATION_CONSISTENCY_CHECK 0x29 FR_STATUS_INITIALIZE_SCHEDULE 0x30 FR_STATUS_ABORT_STARTUP
LOW-WORD of Data[1]	Status Flags: 1 = SESA 2 = SBSA 4 = TCSA 8 = SESB 16 = SBSB 32 = TCSB 64 = Spy 128 = POC CC2 (Cold-start helper POC indicator, if set, event contains the POC state of the cold-start helper) 256 = Trigger Invalid
HI-WORD of Data[1]	Symbol length

4.7.2 Format Description

Element	Type	Format	Remarks
Cycle no.	UInt32	dec/hex	Cycle number
CC-Type.	UInt32	dec	Type of communication controller, see 4.3.3
Sync-State	UInt32	dec	Sync-State: 0 = Not synced passive 1 = Synced active 2 = Not synced
CC-Data[2]	UInt32	dec/hex	Driver flags for internal usage, see 4.7.1
Symbol	UInt32	dec	Collision avoidance symbol, media test symbol wake up symbol: 0 = Void 1 = CAS 2 = MTS 3 = WUS 4 = Network interface doesn't contain symbol information, e.g. if spy-mode is used or the BUSDOCTOR interface.
WakeUp State	UInt32	dec	
Spy flag	UInt32	dec	If 1, async. monitoring has generated this event. Since CANoe 6.1

4.7.3 Example

HEADER	Specific event data
0.003022 Fr SE 0 0 1 3	255 5 2 15 0 0 0 0

4.8 FlexRay Error Event (“EE”)

Description: FlexRay Error event transmitted by the FlexRay hardware.

4.8.1 Controller Specific Information

CC-Type: Cylone I

Field	Description
Data[0]	Error flags from driver API

CC-Type: Cylone II

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Data[0]</td><td>Error packet flag:
0 = No error
1 = FlexCard overflow
2 = PCO error mode changed
3 = Sync frames below minimum
4 = Sync frame overflow
5 = Clock correction failure
6 = Parity error
7 = Receive FIFO overrun
8 = Empty FIFO access
9 = Illegal input buffer access
10 = Illegal output buffer access
11 = Syntax error
12 = Content error
13 = Slot boundary violation
14 = Transmission across boundary
15 = Latest transmit violation</td></tr>
<tr><td>Data[1]</td><td>DWORD layout depends on the error packet value (see previous row)
<table>
<tr><th>Error packet</th><th>Description</th></tr>
<tr><td>2</td><td>0 = Unknown state
1 = FlexRay protocol spec. > CONFIG
2 = FlexRay protocol spec. > NORMAL_ACTIVE
3 = FlexRay protocol spec. > NORMAL_PASSIVE
4 = FlexRay protocol spec. > HALT
5 = FlexRay protocol spec. > READY
6 = FlexRay protocol spec. > STARTUP
7 = FlexRay protocol spec. > WAKEUP</td></tr>
<tr><td>3 or 4</td><td>Bits 0..3 > Sync frames even on channel A
Bits 4..7 > Sync frames even on channel B
Bits 8..11 > Sync frames odd on channel A
Bits 12..15 > Sync frames odd on channel B</td></tr>
<tr><td>5</td><td>Bit 0 > Missing rate correction
Bit 1 > Rate correction limit reached
Bit 2 > Offset correction limit reached
Bit 3 > Missing offset correction
Bit 4..7 > Sync frames even on channel A
Bits 8..11 > Sync frames even on channel B
Bits 12..15 > Sync frames odd on channel A</td></tr>
</table>
</td></tr>
</table>

Field	Description	
		Bits 16..19 > Sync frames odd on channel B
	11.. 15	LOW-WORD of mData[1] > Channel HI-WORD of mData[1] > Slot count

CC-Type: BUSDOCTOR

Field	Description
Data[0]	Error flags from driver API

CC-Type: VN-Interface

Field	Description
Data[0]	Error tag: 0 = FR_ERROR_POC_MODE 1 = FR_ERROR_SYNC_FRAMES_BELOWMIN 2 = FR_ERROR_SYNC_FRAMES_OVERLOAD 3 = FR_ERROR_CLOCK_CORR_FAILURE 4 = FR_ERROR_NIT_FAILURE 5 = FR_ERROR_CC_ERROR 6 = FR_ERROR_OVERFLOW

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Data[1] and Data[2]</td><td>DWORD layout depends on the error tag value (see previous row):
<table>
<tr><th>Error tag</th><th>Value or Bit-Range</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>FR_ERROR_POC_ACTIVE</td></tr>
<tr><td></td><td>1</td><td>FR_ERROR_POC_PASSIVE</td></tr>
<tr><td></td><td>2</td><td>FR_ERROR_POC_COMM_HALT</td></tr>
<tr><td>1 or 2</td><td>Bits 0..3</td><td>Sync frames even on channel A</td></tr>
<tr><td></td><td>Bits 4..7</td><td>Sync frames even on channel B</td></tr>
<tr><td></td><td>Bits 8..11</td><td>Sync frames odd on channel A</td></tr>
<tr><td></td><td>Bits 12..15</td><td>Sync frames odd on channel B</td></tr>
<tr><td>3</td><td>Bit 0</td><td>Missing rate correction</td></tr>
<tr><td></td><td>Bit 1</td><td>Missing rate correction limit reached</td></tr>
<tr><td></td><td>Bit 2</td><td>Offset correction limit reached</td></tr>
<tr><td></td><td>Bit 3</td><td>Missing offset correction</td></tr>
<tr><td></td><td>Bits 4..19</td><td>Clock correction failed counter</td></tr>
<tr><td></td><td>Bit 20..23</td><td>Sync frames even on channel A</td></tr>
<tr><td></td><td>Bit 24..27</td><td>Sync frames even on channel B</td></tr>
<tr><td></td><td>Bit 28..31</td><td>Sync frames odd on channel A</td></tr>
<tr><td></td><td>Bit 32..35</td><td>Sync frames odd on channel B</td></tr>
<tr><td>4</td><td>1</td><td>FR_ERROR_NIT_SENA</td></tr>
<tr><td></td><td>2</td><td>FR_ERROR_NIT_SBNA</td></tr>
<tr><td></td><td>4</td><td>FR_ERROR_NIT_SENB</td></tr>
<tr><td></td><td>8</td><td>FR_ERROR_NIT_SBNB</td></tr>
<tr><td>5</td><td>0x00000001</td><td>POC Error Mode Changed</td></tr>
<tr><td></td><td>0x00000004</td><td>Sync Frames Below Minimum</td></tr>
<tr><td></td><td>0x00000008</td><td>Sync Frame Overflow</td></tr>
<tr><td></td><td>0x00000010</td><td>Clock Correction Failure</td></tr>
<tr><td></td><td>0x00000040</td><td>Parity Error, data from MHDS (internal ERay error)</td></tr>
<tr><td></td><td>0x00000200</td><td>Illegal Input Buffer Access (internal ERay error)</td></tr>
<tr><td></td><td>0x00000400</td><td>Illegal Output Buffer Access (internal ERay error)</td></tr>
<tr><td></td><td>0x00000800</td><td>Message Handler Constraints Flag data from MHDF (internal ERay error)</td></tr>
<tr><td></td><td>0x00010000</td><td>Error Detection on channel A, data from ACS</td></tr>
<tr><td></td><td>0x00020000</td><td>Latest Transmit Violation on channel A</td></tr>
<tr><td></td><td>0x00040000</td><td>Transmit Across Boundary on Channel A</td></tr>
<tr><td></td><td>0x01000000</td><td>Error Detection on channel B, data from ACS</td></tr>
<tr><td></td><td>0x02000000</td><td>Latest Transmit Violation on channel B</td></tr>
<tr><td></td><td>0x04000000</td><td>Transmit Across Boundary on Channel B</td></tr>
</table>
</td></tr>
</table>

4.8.2 Format Description

Element	Type	Format	Remarks
CC-Type.	UInt32	dec/hex	Type of communication controller, see 4.3.3
CC-Data[4]	UInt32	dec/hex	Driver flags for internal usage, see 4.8.1

4.8.3 Example

HEADER	Specific event data
7.344250 Fr EE 0 0 1 2	5 5 65536 0 0

4.9 Example

The following logging file contains FlexRay Receive Message events, FlexRay StartCycle Events and FlexRay Status Event. The decimal number system is used.

```
date Thu May 24 03:16:09 pm 2007
base dec  timestamps absolute
internal events logged
Begin Triggerblock Thu May 24 03:16:09 pm 2007
   0.000000 Start of measurement
   0.000000 Fr SE  0 0 1 3 0  4  2  0  0  0  0
   0.039150 Fr RMSG  0 0 1 1 1 7 Rx 0 3 5  32  96 x  4  0 0  0  0
   0.039150 Fr RMSG  0 0 1 2 1 7 Rx 0 3 5  32  96 x  4  0 0  0  0
   0.039185 Fr RMSG  0 0 1 1 2 7 Rx 0 14 5  32  1193 Message  4  4  10 240  23 134
0  0  0
   ...
   6.484084 Fr EE  0 0 1 2 5  5  65536  0  0
End TriggerBlock
```