

TP/Diagnostics

ASC Logging Format

Specification

Version 1.21 of 2016-09-26

Status	Completed
Publisher	Vector Informatik GmbH © 2016 All rights reserved. Any distribution or copying is subject to prior written approval by Vector. Note: Hardcopy documents are not subject to change management.

Document Management

Revision list

Version	Date	Editor	Section	Changes, comments
1.0	2008-10-27	Trs	All	Creation
1.1	2009-05-18	Trs	1	Added Disclaimer
1.2	2016-01-26	Mar	5	Added Diagnostic Request Interpretation Events
1.2.1	2016-09-26	Mom	all	CI and Layout

Contents

1	Disclaimer	4
2	Overview	4
3	ISO transport protocol observer on CAN	4
3.1	Header	4
3.2	Information as comment	4
3.3	Events	5
3.3.1	Prefix	5
3.3.2	Single Frame	5
3.3.3	First Frame	5
3.3.4	Consecutive Frame	5
3.3.5	Flow Control Frame	5
3.4	Symbols	5
4	Diagnostics requests as macro	6
4.1	Background	6
4.2	Header	6
4.3	Event format	6
4.4	Symbols	6
5	Diagnostic Request Interpretation	7
5.1	Background	7
5.2	Header	7
5.3	Event format	7
5.4	Symbols	7

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

This document specifies the format of transport protocol and diagnostics events in the CANoe/CANalyzer ASC logging.

3 ISO transport protocol observer on CAN

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").

3.2 Information as comment

The ISO TP observer writes TP level information as comments for CAN messages, i.e. CANoe/CANalyzer do NOT process this information in any way when the log file is replayed. Example (lines have been wrapped):

```
date Mon Oct 27 09:35:17 am 2008
base hex timestamps absolute
internal events logged
// version 7.1.0
Begin Triggerblock Mon Oct 27 09:35:17 am 2008
0.000000 Start of measurement
9.078835 1 200 Tx d 8 02 3E 01 00 00 00 00 00
// 1 OTP(05) Atom <tester>->Any_ECU_example: SF Length: 02 [ 3E 01 ]
10.412823 1 200 Tx d 8 02 1A 90 00 00 00 00 00
// 1 OTP(06) Atom <tester>->Any_ECU_example: SF Length: 02 [ 1A 90 ]
10.414203 1 400 Tx d 8 10 0C 5A 90 98 76 54 32
// 1 OTP(07) Atom Any_ECU_example-><tester>: FF Length: 000C [ 5A 90 98 76 54 32 ]
10.416000 1 200 Tx d 8 30 00 14 00 00 00 00 00
// 1 OTP(07) Atom <tester>->Any_ECU_example: FC.CTS: BSmax: 0x00, STmin: 0x14 ms
10.437452 1 400 Tx d 8 21 10 00 99 99 00 01 00
// 1 OTP(07) Atom Any_ECU_example-><tester>: CF Seq.Nr.: 1 [ 10 00 99 99 00 01 00 ]
End Triggerblock
```

3.3 Events

The section lists all ISO TP observer events in CANoe/CANalyzer ASCII logging. See section 3.4 for an explanation of the symbols.

3.3.1 Prefix

Every ISO TP comment is prefixed with basic information:

Format	"// " <CAN-channel> "OTP(" <connection-id> ")" <type> <source> "->" <destination> ":"
Example	// 1 OTP(07) Atom Any_ECU_example-><tester>:

3.3.2 Single Frame

The complete transported data is located within one CAN frame.

Format	"SF Length: " <length> "[" <transported bytes> "]"
Example	SF Length: 02 [1A 90]

3.3.3 First Frame

A data transfer is started. The number of bytes to be sent is indicated.

Format	"FF Length: " <length> "[" <transported bytes> "]"
Example	FF Length: 000C [5A 90 98 76 54 32]

3.3.4 Consecutive Frame

Further data as part of the transfer.

Format	"CF Seg.Nr.: " <SN> "[" <transported bytes> "]"
Example	CF Seq.Nr.: 1 [10 00 99 99 00 01 00]

3.3.5 Flow Control Frame

The data flow is regulated with this type of frames.

Format	"FC." <FC-type> ": BSmax: " <BS> ", STmin: " <STmin> "ms"
Example	FC.CTS: BSmax: 0x00, STmin: 0x14 ms

3.4 Symbols

Explanation of symbols used in descriptions above.

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example (hex)	Special
<CAN-channel>	-	>=1	CAN channel	1..32	1	
<connection-id>	2	-	Identify data transfer	01..63	07	

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example (hex)	Special
<type>	4/5	4/5	Event type	"Info", "Warn", "Error", "Atom", "Data"	"Atom"	
<source>	>=1	>=1	Source node	Numeric or symbolic	"Any_ecu_example"	
<destination>	>=1	>=1	Destination node	Numeric or symbolic	"<tester>"	
<length>	2-4	2-3	Number of bytes transferred	0...0xFF	000C	
<transported bytes>	>= 6	>=7	Bytes transported within this CAN message	...	[01 02]	
<SN>	4	3	Sequence number	0...0xF	1	
<FC-type>	2	5	Flow Control frame type	"CTS", "WT", "OVFLW"	"CTS"	"FC.Illegal Flow Status"
<BS>	4	3	Block size	0..0xFF	0x00	
<STmin>	4	3	Minimum separation time	0..0xFF	10	

4 Diagnostics requests as macro

4.1 Background

For macro recording, diagnostics requests sent from the diagnostics console or the fault memory window are stored in an ASCII logging format that can be replayed in CANoe.

Note: The responses are NOT recorded, since they should be generated ad hoc by the diagnostics server (ECU) in the first place.

4.2 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").

4.3 Event format

Format	<timestamp> "DiagRequest[" <ECU-qualifier> "]" <byte-sequence>
Example	1.765500 DiagRequest[Any_ECU_example] 1A 90

4.4 Symbols

Explanation of symbols used in descriptions above.

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example (hex)	Special
<timestamp>	-	>=5	Standard trace time stamp	...	1.234	
<ECU-qualifier>	>=1	>=1	Unique identifier of the diagnostics description	-	Any_ECU_example	
<byte-sequence>	>=2	-	Representation of the bytes sent to the ECU by the tester	-	1A 90	

5 Diagnostic Request Interpretation

5.1 Background

For diagnostic requests sent by CANoe, the target ECU, the active diagnostic variant and the used diagnostic service are logged.

This information ensures that the request and response are interpreted correctly in CANoe.

5.2 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").

5.3 Event format

Format	<timestamp> Diag: [<EcuID>,<VariantID>,<ServiceID>] Request <ServiceQualifier> sent to <EcuQualifier> (Variant: <VariantQualifier>)
Example	4.724643 Diag: [00000001,00000000,00000000] Request DefaultSession_Start sent to Door (Variant: CommonDiagnostics)

5.4 Symbols

Explanation of symbols used in descriptions above.

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example (hex)	Special
<timestamp>	-	>=5	Standard trace time stamp	...	1.234	
<EcuID>	8	8	Internal ID of the ECU (diagnostic description)	00000000 – FFFFFFFF	00000007	
<VariantID>	8	8	Internal ID of the active diagnostic variant	00000000 – FFFFFFFF	00000007	
<ServiceID>	8	8	Internal ID of the diagnostic service	00000000 – FFFFFFFF	00000007	
<EcuQualifier>	>=1	>=1	Unique identifier of the diagnostic description	-	Any_ECU_example	

<VariantQualifier>	>=1	>=1	Unique identifier of the diagnostic variant	-	BaseVariant	
<ServiceQualifier>	>=1	>=1	Unique identifier of the diagnostic service	-	DevelopmentData_Read	