Document Title	Requirements on Log and Trace
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	350
Document Status	Final
Part of AUTOSAR Standard	Foundation
Part of Standard Release	1.3.0

Document Change History

Date	Release	Changed by	Change Description
2017-12-08	1.3.0	AUTOSAR Release Management	• No content changes
2017-10-27	1.2.0	AUTOSAR Release Management	• --Migration of document to standard "Foundation"--
2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Renamed "RS_BSWAndRTEFeatures" into "RS_Features • Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• TPS_STDT_0078 formatting • Traceability of BSWAndRTE_Features
2010-09-30	3.1.5	AUTOSAR Administration	• Initial release

Disclaimer

Table of Contents

1 Scope of Document

This following specification defines the functional and non-functional requirements of the BSW module Diagnostic Log and Trace (LT) for AUTOSAR Release 4.0.

This BSW component LT is new to AUTOSAR since Release 4.0. The existing component DET provides only a minimal interface for Error Tracing and do not specify how trace data is stored.

The focus of this document is to specify the requirements for:

- The interface of LT to other BSW modules
- The interface to RTE/VFB Tracing
- The interface to Applications
- The transmission format of the log and trace messages
- The storage format of the log and trace messages
- The internal interface to the LT communication module
- The configuration of LT

The focus is NOT to specify:

- The transport layer of the communication over the LT communication module

2 Conventions to be used

- The representation of requirements in AUTOSAR documents follows the table specified in [5].

- In requirements, the following specific semantics shall be used (based on the Internet Engineering Task Force IETF).

  The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as:

  - SHALL: This word means that the definition is an absolute requirement of the specification.
  - SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
  - MUST: This word means that the definition is an absolute requirement of the specification due to legal issues.
  - MUST NOT: This phrase means that the definition is an absolute prohibition of the specification due to legal constraints.
  - SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
  - SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
  - MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

3 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
Log and trace message	A log and trace message contains all data and options to specify a log and trace event in a software.
User	The user of LT is the programmer of the software, which uses the LT API to generate log and trace messages.
Log	The user generates log messages on demand. Each time the user wants to show some information about state changes or value changes, he adds an API call to LT.
Trace	Trace messages can be generated by instrumentation of the code (e.g. VFB traces). The instrumented code calls the API of LT.
ECU ID	ECU ID is the name of each ECU.
Session ID	Session ID is the identification number of a log or trace session. If an Application is instantiated several times the log sessions get a new Session ID. A Application can have several log or trace sessions. A BSW module uses the module-number as Session ID.
Application ID	Application ID is a short name of the Application/BSW module. It identifies the Application/BSW module in the log and trace message.
Context ID	Context ID is a user defined ID to group log and trace messages produced by an Application/BSW module to distinguish functionality. Each Application ID can own several Context IDs. Context ID's are grouped by Application ID's. Context IDs shall be unique within an Application ID. The identification of the source of a log and trace message is done with a pair of Application ID and Context ID.
Message ID	Messaged ID is the ID to characterize the information, which is transported by the message itself. A Message ID identifies a log or trace message uniquely. It can be used for identifying the source (in source code) of a message and it can be used for characterizing the payload of a message.
Log and trace level	A log level defines a classification for the severity grade of a log message. The trace status provides information if a trace message should be send.

Time	Each log and trace message may contain a time attribute. The time attribute is a free defined time-value. It is the time since the start of the ECU.
External client	An external client is a tool, which can be run on a PC or another ECU, which is connected to LT over DCM or over the LT communication module.

4 Functional Overview

LT provides a generic Logging and Tracing functionality for Application's and the BSW modules RTE, DET and DEM. Main focuses of this document are to specify the container, how data is buffered locally and exported over a communication interface. The following figure shows, how LT is integrated into the AUTOSAR architecture and how the main functionality of LT shall be realized.

LT provides the following functionalities:

- Logging
  - Logging of errors, warnings and info messages from AUTOSAR Application's, providing a standardized AUTOSAR interface
  - Gather all log and trace messages from all AUTOSAR Application's in a centralized AUTOSAR service component (LT) in BSW
  - Log messages from DET (Classic Platform)
  - Log messages from DEM (Classic Platform)
- Tracing
  - Trace RTE/VFB
- Control
  - Enable/disable individual log and trace messages
  - Control trace levels individually by back channel
- Generic
  - LT available during debugging and production phase
  - Access over standard diagnosis or platform specific test interface
  - Security mechanisms to prevent misuse in production phase

5 Requirements Specification

5.1 Requirements Tracing

Requirement	Description	Satisfied by
RS_BRF_01016	-	RS_LT_00042
RS_BRF_01024	-	RS_LT_00022, RS_LT_00027
RS_BRF_01056	-	RS_LT_00034, RS_LT_00035
RS_BRF_01240	-	RS_LT_00013, RS_LT_00014
RS_BRF_01440	-	RS_LT_00001, RS_LT_00003, RS_LT_00004, RS_LT_00005, RS_LT_00006, RS_LT_00007, RS_LT_00008, RS_LT_00009, RS_LT_00010, RS_LT_00011, RS_LT_00012, RS_LT_00013, RS_LT_00014, RS_LT_00016, RS_LT_00017, RS_LT_00018, RS_LT_00019, RS_LT_00020, RS_LT_00021, RS_LT_00023, RS_LT_00025, RS_LT_00030, RS_LT_00032, RS_LT_00034, RS_LT_00035, RS_LT_00036, RS_LT_00038, RS_LT_00039, RS_LT_00040, RS_LT_00041
RS_BRF_01624	-	RS_LT_00016
RS_BRF_01824	-	RS_LT_00039
RS_BRF_02008	-	RS_LT_00029
RS_BRF_02144	-	RS_LT_00003, RS_LT_00035
RS_BRF_02160	-	RS_LT_00010, RS_LT_00028
RS_BRF_02168	-	RS_LT_00033
RS_BRF_02184	-	RS_LT_00037, RS_LT_00038
RS_BRF_02200	-	RS_LT_00005, RS_LT_00031
RS_BRF_02208	-	RS_LT_00029
RS_Main_00060	AUTOSAR shall provide a standardized software interface for communication between Applications	RS_LT_00043, RS_LT_00044, RS_LT_00045, RS_LT_00046, RS_LT_00051, RS_LT_00052
RS_Main_00400	AUTOSAR shall provide a layered software architecture	RS_LT_00047, RS_LT_00048, RS_LT_00049, RS_LT_00050

5.2 Functional Requirements

5.2.1 Log and trace interfaces

5.2.1.1 Generic

5.2.1.1.1 [RS_LT_00047] Initialization and registration.

⌈

Type:	Valid
Description:	Logging shall support to initialize the logging framework, and to register the source of logging information.
Rationale:	To be able to filter and associate logging information with the origin, it is necessary that Applications register themselves at the logging framework.
Use Case:	Associate logging information with the origin, apply filter settings, and provide additional information.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00400)

5.2.1.1.2 [RS_LT_00048] Meta information about Applications.

[

Type:	Valid
Description:	Logging shall enable Applications to provide additional information about themselves.
Rationale:	Being able to identify the origin of generated logging information.
Use Case:	Sort, filter and associate received logging information.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00400)

5.2.1.1.3 [RS_LT_00049] Providing Logging Information.

[

Type:	Valid
Description:	Logging shall enable Applications to provide logging information. The logging information shall include information about its severity.
Rationale:	While developing applications, it is important to get additional information what is going on internally of an application.
Use Case:	Getting internal information of an application, e.g.: Variable values, the current internal state of a state machine, and other information.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00400)

5.2.1.1.4 [RS_LT_00050] Grouping of Logging Information.

[

Type:	Valid
Description:	Logging shall support to logically group logging information.
Rationale:	Cluster logging information, which logically belong together.
Use Case:	Associate or filter all logging information which belong together.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00400)

5.2.1.1.5 [RS_LT_00051] Logging Information targets.

[

Type:	Valid
Description:	Logging shall allow to select the destination of the provided logging information.

Rationale:	There are different possibilities where the logging information can be stored.
Use Case:	Forward logging information to the console, to the file system, or to send it via the communication bus.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.1.6 [RS_LT_00052] Early logging.

[

Type:	Valid
Description:	Logging framework shall provide early logging possibility. The API shall be able to handle log message attempts as well as creating new log contexts instances, before the main initialization phase of the Logging back-end was accomplished.
Rationale:	Calling the initialization procedure at earliest is possible when the program code is executed within the main event loop. Before that, applications might create global/static objects which constructors might already log useful information. Since global/static object are instantiated before the main event loop enters, the Logging framework **must** support calls to all its interfaces without crashing. Ideally it **shall** buffer log attempts (an unspecified amount) and statically created log context instances and process them ASAP after the initialization phase.
Use Case:	Create log contexts and/or initiate log messages before initialization phase of the Logging back-end was accomplished.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.1.7 [RS_LT_00001] The LT shall transmit log and trace messages from several sources over a communication interface to a receiving external client.

[

Type:	Valid
Description:	The LT module shall be a BSW module. It shall receive log and trace messages from several sources, like Applications, some BSW modules and the RTE. These messages may be transmitted over a communication interface to an external client. Log and trace is a debugging mechanism needed by a lot of ECU's. A log and trace module shall provide a mechanism to gather log and trace messages from several sources. The log and trace module buffers the log and trace entries if necessary and shall have a connection to an external client.
Rationale:	Each Tier1 uses its own mechanisms to provide such a logging interface, using some internal or external debugging interfaces. The format of the logging content also differs from ECU to ECU. When testing several ECU's, many different tools and parsers are needed to get the right information out of the logs. A standard Diagnostic Logging Component with standardized logging content may help to reduce the testing efforts and enable new automated testing mechanisms. Also the number of tools could be reduced by a standard logging content and protocol.
Use Case:	• Development support • Functional Testing

	• Test Automation • Test against models • Driver intensive tests • Advanced Diagnostic Tracing, optional over telematic services
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.1.8 [RS_LT_00002] All log and trace messages sent by an ECU shall have a standardized transmission format and a standardized storage format.

[

Type	Valid
Description:	A specified format shall be defined, which covers all requirements of log and trace. Some examples of the information stored in a log or trace message are the source, the context and the timestamp of the message to be able to filter the log and trace messages.
Rationale:	Since logging and tracing is an important mechanism for testability and proofing product quality, it is necessary to standardize the transmitted and stored data format. This is important for archiving, comparing and analyzing of log or trace messages. Also it may be possible to build common tools to interpret the incoming data.
Use Case:	• Applications sends a log message • LT sends the message over an interface to a data storing external client • The stored data of different ECU's are interpreted by the external client • Log and trace messages from different ECU's can be merged to understand relationship of behavior from distributed functionality
Applies to:	FO
Dependencies:	--
Supporting Material:	see [1]

]()

5.2.1.2 Interface for Applications

5.2.1.2.1 [RS_LT_00003] Applications shall have the possibility to send log or trace messages to the LT module.

[

Type:	Valid
Description:	LT shall offer a generic interface for Applications independent from the type of log and trace message.
Rationale:	To reduce the amount of interfaces a generic and message independent interface may be implemented.
Use Case:	• Interfacing Application with LT module • Tracing or logging messages from Applications
Applies to:	FO
Dependencies:	--
Supporting Material:	see [1]

](RS_BRF_02144,RS_BRF_01440)

5.2.1.2.2 [RS_LT_00043] Interface for logging information.

[

Type:	Valid
Description:	Logging shall provide Interfaces for applications to forward its logging information, including the associated severity level.
Rationale:	Forward logging information to the logging framework.
Use Case:	Forward logging information to a logging framework for further process it (e.g. store it locally, or forward it to the communication bus).
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.2.3 [RS_LT_00044] Provide raw buffer content.

[

Type:	Valid
Description:	Logging shall support to provide the content of a raw buffer as logging information.
Rationale:	The original content of a buffer might be of interest for developing purpose.
Use Case:	Get the exact raw data which is located in the memory.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.2.4 [RS_LT_00045] Check the current severity level.

[

Type:	Valid
Description:	Logging shall provide the possibility for applications to check the current active severity level.
Rationale:	To avoid unnecessary CPU and/or memory consumption, which is needed for the generation of logging information, Applications shall have the possibility to first check whether or not its created logging information will be filtered out anyway by the underlying logging framework.
Use Case:	Avoidance of unnecessary CPU and/or memory consumption.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.2.5 [RS_LT_00046] Conversion functions for hexadecimal and binary values.

[

Type:	Valid
Description:	Logging shall provide the possibility for applications to convert decimal values into the hexadecimal or binary system.
Rationale:	Provide some debug information as a hexadecimal or as a binary value, instead of a decimal value.
Use Case:	Providing logging information as hexadecimal value or as binary value.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_Main_00060)

5.2.1.2.6 [RS_LT_00004] The LT shall provide the actual set of log levels and the trace status to an Application.

[

Type:	valid
Description:	Application shall be aware of its log levels and trace status in order not to generate unnecessary log and trace messages. Log level and trace status should be checked by the Applications before sending the messages.
Rationale:	The generation of unnecessary load on the communication interface and for the system has to be avoided. The LT shall provide information about the actual configuration.
Use Case:	• Control of the log levels and of the trace status
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.2.7 [RS_LT_00005] For each Application the interface to LT shall be configured.

[

Type:	valid
Description:	The generation of the LT interface of each Application shall be enabled by the configuration.
Rationale:	Only the configured interfaces to the Applications, which are using log and trace, shall be generated.
Use Case:	
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02200,RS_BRF_01440)

5.2.1.3 DET trace interface

5.2.1.3.1 [RS_LT_00006] Trace events from errors generated by BSW and Applications shall be forwarded to the LT module

[

Type	Valid
Description:	The DET receives trace events from errors generated by BSW and Applications during debugging time. These events may be forwarded to the LT module.
Rationale:	To have an overview of all log, trace and error messages and to set all of them in the correct context, it is important to have all these messages and events in one list. Also it is not practicable to use more than one mechanism to report errors, logs and traces to an external client. So all these sources may be forwarded to the LT module.
Use Case:	• In a debugging scenario, an Application or BSW Module uses the DET interface to trace an error. This error is forwarded by the DET module to the LT module. The LT turns these events into a log message and sends it to the external client
Applies to:	CP
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.4 DEM trace interface

5.2.1.4.1 [RS_LT_00007] The DEM shall forward error events to the LT module

[

Type	Valid
Description:	The DEM may forward error events to the LT module.
Rationale:	It may be possible to get an overview of all error messages in an ECU. It may be possible to set them in the correct context with the error events reported by DEM. This makes an analysis of the reported errors more efficient and gives a correct picture of a failure situation.
Use Case:	• A Application or BSW module sets an DTC in the DEM. The DEM forwards this event to the LT. The LT turns these events in the LT format and sends it over a communication interface to a LT external client.
Applies to:	CP
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.5 RTE/VFB trace interface

5.2.1.5.1 [RS_LT_00008] RTE shall provide an interface for LT to trace RTE/VFB calls.

[

Type	Valid
Description:	RTE shall provide the possibility to trace the VFB.
Rationale:	In the future more and more Applications will be integrated in one ECU. As a consequence the communication between Application is done locally and not over an external traceable bus like CAN or Flexray. It is important to trace the internal communication over RTE/VFB.
Use Case:	• Trace of VFB interface • Access to VFB for advanced diagnostic services, see RS_BRF_00224
Applies to:	CP
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.5.2 [RS_LT_00009] The LT shall implement an interface to trace the RTE/VFB.

[

Type:	valid
Description:	VFB-Tracing shall be provided to the RTE generation system. The LT module shall implement the handling of the RTE/VFB trace events.
Rationale:	To understand the communication between the Applications it is important to trace all function calls to the Applications except the RTE system function calls.
Use Case:	--
Applies to:	CP
Dependencies:	--

Supporting Material:	--

](RS_BRF_01440)

5.2.1.5.3 [RS_LT_00010] A global switch shall be defined to switch on and off the RTE tracing.

[

Type:	valid
Description:	There shall be a global switch to turn on and off the RTE tracing of LT at configuration time and at run time.
Rationale:	To have a high level activation for all RTE tracing messages a global switch is needed.
Use Case:	• Turning all tracing events for LT off • Turning all tracing events for LT on
Applies to:	CP
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440,RS_BRF_02160)

5.2.1.5.4 [RS_LT_00011] The LT shall implement the handling of the RTE/VFB trace events

[

Type:	Valid
Description:	The LT may implement the handling of the RTE/VFB trace events. The groups may be configured at configuration time and the events may be assigned to one group. At run time the log level of each group can be set individually.
Rationale:	To understand the communication between the Applications it is important to trace the function calls of an Application.
Use Case:	--
Applies to:	CP
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.1.5.5 [RS_LT_00012] LT shall provide a solution to trace events linked to implicit communication mechanism

[

Type:	Valid
Description:	LT may provide a solution to trace events linked to implicit communication mechanism. The implicit communication may be traced at runnable invocation and termination.
Rationale:	From the point of view of an Application no extra routine for receiving or sending signals (Sender Receiver Communication) with implicit sending/receiving is called. The RTE manages the copying of the data before invocation and after exiting.
Use Case:	• Tracing implicit communication of Applications
Applies to:	CP
Applies to:	
Dependencies:	--

Supporting Material:	--

](RS_BRF_01440)

5.2.2 Format of log and trace message

5.2.2.1 [RS_LT_00013] The transmitted data shall be packetized.

[

Type:	Valid
Description:	The log and trace messages shall be packet oriented. Each packet shall have one or several headers and a payload. The headers may be used for identifying, filtering and classifying the log and trace message. The payload shall contain the parameters given from the Application or BSW.
Rationale:	LT shall not care about the under laying transport protocol. LT shall define the log or trace message format.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440,RS_BRF_01240)

5.2.2.2 [RS_LT_00014] The transport format shall be binary.

[

Type:	Valid
Description:	The log and trace messages shall be in binary format. This is necessary for reducing the bandwidth and the CPU consumption.
Rationale:	Scope of the LT component is to log and trace events without perturbing the ECU behavior. CPU consumption and bandwidth occupation may be reduced by using a binary format for the protocol.
Use Case:	--
Applies to:	FO
Dependencies:	--
Conflicts	None
Supporting Material	--

](RS_BRF_01440,RS_BRF_01240)

5.2.2.3 [RS_LT_00016] The format shall deal with Big and Little Endianess.

[

Type:	Valid
Description:	The Endian format shall be considered in the message specification. The message shall signal which Endianess mode is actually used.
Rationale:	Big and Little Endianess of data representation has mixed occurrences in automotive ECUs. The message format shall be able to deal with both. To reduce CPU load it shall be possible to transport the data in the corresponding format of the ECU.
Use Case:	• Marker for format
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440,RS_BRF_01624)

5.2.2.4 [RS_LT_00017] Each log and trace message shall contain a timestamp, which will be added to the message during reception of the message in the LT module

[

Type:	Valid
Description:	Each log and trace message may contain a timestamp, which shall be added to the message during reception of the message in the LT module.
Rationale:	For comparing and analyzing complex systems a timestamp is important. The time shall be in a defined granularity.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.5 [RS_LT_00018] A global message counter shall be implemented, to detect messages loss.

[

Type:	Valid
Description:	Every time a message from an Application is received by the LT module or other software module, it shall increment its global messages counter and set the counter in the message.
Rationale:	It is important to know if a message is lost. A receiving external client may then detect if messages are lost (e.g.because of a buffer overflow in LT or network loss).
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.6 [RS_LT_00019] For each log message, a log level shall be provided.

[

Type:	Valid
Description:	Log levels shall be for example error, fatal, debug, information or verbose.
Rationale:	Standard for logging tools
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.7 [RS_LT_00020] The log and trace message shall contain a parameter, which represents the source of the log and trace message

[

Type:	Valid
Description:	The log and trace message may contain a parameter, which represents the source of the log and trace message.

Rationale:	The source can be identified by the Session ID of the port interface (port defined argument value) which was used to send the message from Application to LT.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.8 [RS_LT_00021] There shall be a logical grouping for log messages by using different identifiers.

[

Type:	Valid
Description:	There shall be additional parameters describing the group of the log and trace message.
Rationale:	Identifying some logically groups of log and trace to define some filtering action on the external client.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.9 [RS_LT_00022] Each ECU shall have its unique ECU ID

[

Type:	Valid
Description:	Each ECU may have its unique ECU ID, which may be a parameter of each log and trace message.
Rationale:	An external client receives tracing from different ECU's so it's important to know the source.
Use Case:	• Distinguish tracing from different ECU's
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01024)

5.2.2.10 [RS_LT_00023] The payload shall transport the parameters of a log and trace message.

[

Type:	valid
Description:	In the payload the parameters of each log and trace message from the Application shall be transported. It may be possible to transport more than one parameter.
Rationale:	--
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.11 [RS_LT_00024] It shall be possible to transmit the parameters in a raw format.

[

Type:	Valid
Description:	In most cases the parameter shall be transmitted in the raw format. This means that a separate file shall be provided containing the information about the transmitted parameters.
Rationale:	The non-verbose mode is for saving memory and bandwidth, because on the ECU only the ID's of the log and trace messages shall be stored.
Use Case:	• Interpreting binary data
Applies to:	FO
Dependencies:	--
Supporting Material:	--

]()

5.2.2.12 [RS_LT_00044] There shall be the possibility to transmit the parameters with additional information about themselves (self-description).

[

Type:	valid
Description:	The log and trace messages shall contain a description of the parameters in verbose mode. No additional description file needs to be provided in this case.
Rationale:	In the verbose mode all parameters of the payload contain a self-description of their own type (e.g. integer/float, bit length) and some additional information (e.g. variable name, unit, etc).
Use Case:	• Interpreting binary data
Applies to:	FO
Dependencies:	--
Supporting Material:	--

]()

5.2.2.13 [RS_LT_00025] It shall be possible to transmit ASCII text in log or trace messages.

[

Type:	Valid
Description:	Because log and trace messages often consists of texts, there shall be the possibility to transmit text strings. It may be possible to replace the given text by an ID which may be replaced by the text in the external client (Compare verbose – non-verbose mode).
Rationale:	In small ECU strings can not be stored because of their memory usage. For this reasons these strings may be replaced by a message ID.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.14 [RS_LT_00026] The data in non-verbose mode shall be described by an extra file

[

Type:	valid
Description:	The data in non-verbose mode need to be described by an extra file. This file may be in a XML format.
Rationale:	In small ECUs memory is very limited, so temporarily buffered data shall be very small.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.2.15 [RS_LT_00027] Each message shall have a unique identifier significant for identifying the source of the tracing.

[

Type:	Valid
Description:	Each message may have a unique identifier significant for identifying the source of the tracing. The ID of a specific message shall be permanent over the developing phase and shall be unique for a given ECU.
Rationale:	Easy identifying the right information in the log and trace message is very important. Unique message IDs helps doing this.
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01024)

5.2.3 Transport interfaces

5.2.3.1 Generic

5.2.3.1.1 [RS_LT_00028] A control message shall be implemented to permit the external client to evaluate the round trip time.

[

Type:	valid
Description:	A control message shall be implemented to permit the external client to evaluate the round trip time. The external client shall use this message to synchronize the individual log and trace messages from the different ECUs.
Rationale:	The log and trace time information may enable a user to understand what happens if there is a correlation among the recorded events. It is important to be able to refer all the events logged to an absolute time scale.
Use Case:	• An external client is able to synchronize packet reception from different sources • A round trip time calculation for each ECU is done • In each message the ECUs local time is transmitted • For every message from several ECUs a global time line can be calculated
Applies to:	FO
Dependencies:	--

Supporting Material:	--

](RS_BRF_02160)

5.2.3.1.2 [RS_LT_00029] A protection against unauthorized access in production phase shall be provided.

[

Type:	valid
Description:	In production phase, the communication module (see RS_LT_00034) shall be disabled by default. In production phase LT shall use the security mechanisms provided by DCM to handle the access to the log and trace messages. A mechanism shall be implemented to enable the communication module for testing phase permanently.
Rationale:	Only authorized accesses to LT shall be allowed during production phase to avoid misuse.
Use Case:	• Avoid LT misuse
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02008,RS_BRF_02208)

5.2.3.1.3 [RS_LT_00030] Monitoring and shaping of LT log and trace event amount

[

Type:	Valid
Description:	LT may shape the traffic bandwidth, depending on which transport interface is used for communication. The traffic shaping shall be done at runtime.
Rationale:	Since LT shall be used also in production phase, it is important that the traffic produced by the LT component will not cause an overload of the network and the ECU resources.
Use Case:	• Bandwidth occupation has to be controlled • Bandwidth configuration has to be permitted
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.3.1.4 [RS_LT_00031] The LT shall be configurable at runtime.

[

Type:	Valid
Description:	LT shall be configurable at runtime. LT shall have for this propose runtime configurable variables which are initialized with the corresponding configuration parameters. It shall be possible to query the LT runtime configuration. The configuration variables shall be writeable and readable over with some control .mechanism triggered remotely.
Rationale:	To adapt LT behavior to ECU and network resources it is important to have the capability of configuring it.
Use Case:	• Changing default levels with storing in NVRAM • Query LT state
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02200)

5.2.3.1.5 [RS_LT_00032] A protocol shall be implemented to be able to set and query the trace status and log levels of log and trace sources of each ECU.

[

Type:	valid
Description:	LT shall support control messages to set the trace status and log level of each source of each ECU.
Rationale:	--
Use Case:	• Enabling/disabling trace and log messages at different levels. • Set the level of the log messages generated by different sources of each ECU.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.3.1.6 [RS_LT_00033] A list of all log and trace sources of an ECU shall be accessible from the external client.

[

Type:	Valid
Description:	It shall be possible to get a list of all Application IDs and context IDs with corresponding trace status and log levels which are registered to the LT module.Each Application or BSW module which wants to produce trace or log messages shall register to the LT module.
Rationale:	Users need to know what information can be traced or logged in order to monitor ECU activity.
Use Case:	• Log messages from different ECU's can be merged to understand relationship of behavior from distributed Applications. • Set the level of the tracing generated by different sources of each ECU.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02168)

5.2.3.2 Communication interface

5.2.3.2.1 [RS_LT_00034] LT shall support a generic API for communicating over a LT communication module.

[

Type:	valid
Description:	At debugging phase a communication interface with a high bandwidth is needed. This communication interface is out of scope of the AUTOSAR specification. To use this communication interface with LT a generic API shall be defined. The user shall be able to implement a board specific communication module to communicate over a board specific communication interface.
Rationale:	--
Use Case:	--
Applies to:	FO
Dependencies:	--

Supporting Material:	--

](RS_BRF_01440,RS_BRF_01056)

5.2.3.3 DCM transport interface

5.2.3.3.1 [RS_LT_00035] The DCM shall provide an interface for LT to transport log and trace messages over a diagnostic session.

[

Type:	Valid
Description:	DCM shall provide an interface for LT to send and receive data over the diagnostic service. Log and trace messages are sent over this service and control messages for LT are received. Because log and trace messages are event triggered and the storage on the ECU is limited, these messages shall be sent when they occur. As diagnostic services can only send a small amount of data, the amount of data shall be limited, see RS_LT_00030.
Rationale:	LT needs an interface to send log and trace message out of the ECU. DCM provides a bus independent access to the ECU over standardized diagnostics. This is available during production phase and provides a secured session control. As diagnostic services can only send a small amount of data, the amount of data shall be limited, see RS_LT_00030. For higher bandwidth the communication module shall be used, see RS_LT_00034.
Use Case:	• Transmitting log and trace message during a diagnostic session • Advanced Diagnostic Tracing, optional over telematic services
Applies to:	CP
Dependencies:	RS_LT_00030, RS_LT_00034
Supporting Material:	--

](RS_BRF_02144,RS_BRF_01440,RS_BRF_01056)

5.2.4 Operational function

5.2.4.1 Initialization and shutdown

5.2.4.1.1 [RS_LT_00036] The LT shall provide a buffer for storing log and trace messages before initialization

[

Type:	Valid
Description:	The LT may provide a buffer for storing log and trace messages. This may be a standard C-initialized buffer which shall be read out after init of LT.
Rationale:	This requirement shall guarantee that the LT module can provide its services to other BSW modules also before the complete initialization of all BSW modules.
Use Case:	• Logging is available at startup phase • Logging is available during shutdown process
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.2.4.2 Normal operation

5.2.4.2.1 [RS_LT_00037] There shall be a buffer to store log and trace message locally.

[

Type:	Valid
Description:	If the communication module is not enabled or no external client for dumping log and trace message is connected, log and trace messages shall be stored locally in a memory buffer. The size of the buffer shall be set at configuration time.
Rationale:	To prevent a lack of communication e.g. at startup or if no external client is connected, the LT shall store some messages to be dumped as soon as communication channel is available.
Use Case:	• ECU startup • No communication available
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02184)

5.2.4.2.2 [RS_LT_00038] A mechanism shall be implemented to be able to set the trace status and log levels of registered Application IDs and context IDs of each Application.

[

Type:	valid
Description:	LT shall permit to turn off or on traces or to set log levels for registered Application IDs and context IDs at runtime.
Rationale:	To gain efficiency in bandwidth and ECU resource usage a mechanism shall be provided to tune the transmission from Applications to LT. The LT module shall provide some control variables which represent the actual log levels and trace status of the Applications and contexts.
Use Case:	• Enabling/disabling trace status and log levels.
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_02184,RS_BRF_01440)

5.2.4.2.3 [RS_LT_00039] The LT shall provide the possibility to store configuration data in a persistent way.

[

Type:	Valid
Description:	Some of the log and trace configurations like specific log levels shall be stored persistent.
Rationale:	The LT can be configured at runtime. Some of this configuration data has to be stored to guarantee that at every startup the configuration would be the same.
Use Case:	---
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440,RS_BRF_01824)

5.2.4.2.4 [RS_LT_00040] the LT component shall be able to filter log and trace messages

[

Type:	valid
Description:	In order to adapt network traffic to bandwidth availability, the LT component may be able to filter log and trace messages. A policy shall be defined to distinguish messages by different log and trace levels. The configuration of the filter level may be different for storing log and trace messages temporarily and communication over DCM or the LT communication module.
Rationale:	--
Use Case:	--
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.3 Non functional Requirements

5.3.1 [RS_LT_00041] LT shall be a central software component for the log and trace functionality.

[

Type:	valid
Description:	Only one SW module per ECU shall gather the log and trace messages centrally.
Rationale:	None
Use Case:	None
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01440)

5.3.2 [RS_LT_00042] The Log and trace SW component shall be part of the system during production phase

[

Type:	valid
Description:	The Log and trace SW component may be also part of the system during production phase. It is needed for use cases like test drives and remote diagnosis.
Rationale:	None
Use Case:	• Test drives • Advanced remote diagnosis
Applies to:	FO
Dependencies:	--
Supporting Material:	--

](RS_BRF_01016)

6 References

[1] Requirements on AUTOSAR Features
AUTOSAR_RS_Features.pdf

[2] Specification of RTE,
AUTOSAR_SWS_RTE.pdf

[3] Layered Software Architecture,
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf.pdf

[4] Specification of Development Error Trace,
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[5] Software Standardization Template
AUTOSAR_TPS_StandardizationTemplate.pdf