Elektrobit

EB tresos® AutoCore Generic 8 DLT documentation

release notes update for the Dlt module

product release 8.8.7

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `Dlt` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

Release notes details

- EB tresos AutoCore release version: 8.8.7
- EB tresos Studio release version: 29.2.1
- AUTOSAR R4.2 Rev 1
- Build number: B619567

2. Dlt module release notes

- AUTOSAR R4.2 Rev 1
- AUTOSAR SWS document version: 4.2.1
- Module version: 1.8.11.B619567
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 1.8.11

2023-03-03

- ASCDLT-1015 Fixed known issue: Dlt filters messages wrongly with BSW distribution enabled

Module version 1.8.10

2022-10-12

- ASCDLT-974 Fixed known issue: Dlt may produce array bounds overflow when storing log channel names
- ASCDLT-967 Fixed known issue: Dlt does not send log messages for an ApplicationId/ContextId tuple with a log level higher than the channel log level
- ASCDLT-978 Fixed known issue: Dlt may discard terminating null character when storing EcuId name or context description for VFB functions

Module version 1.8.9

2022-07-04

- ASCDLT-947 Fixed known issue: Dlt fails to generate when DltRteUsage is enabled but no Rte port for a Dlt interface is enabled
- ASCDLT-957 Fixed known issue: Dlt might calculate an incorrect internal ApplicationId counter that leads to an out-of-bounds ROM access

Module version 1.8.8

2022-03-09

- ASCDLT-926 Fixed known issue: Dlt might send wrong data for BufferSize in Dlt_ComCopyRxData() API

Module version 1.8.7

2021-10-27

- Internal module improvement. This module version update does not affect module functionality.

Module version 1.8.6

2021-06-25

- ASCDLT-887 Fixed known issue: Dlt might refuse log/trace messages if AUTOSAR 4.3.1 is used and Message Options bitfield is defined according to the AUTOSAR 4.3.1 format
- ASCDLT-894 Fixed known issue: Dlt might have unexpected behavior when control messages are used and nested critical sections are not considered

Module version 1.8.5

2021-03-05

- ASCDLT-848 Fixed known issue: Enabling DltEnableBswDistribution and DltImplementVfbTrace can lead to compiler warnings and/or errors
- ASCDLT-858 Fixed known issue: Dlt returns a positive response for a GetLogInfo control message request with an invalid tuple
- ASCDLT-876 Fixed known issue: Dlt might send log/trace messages on wrong log channel(s) when pre-compile time configuration of log channel assignments is used

Module version 1.8.4

2020-10-23

- ASCDLT-820 Fixed known issue: The Dlt has an increased memory usage when RteVfbTraceEnabled is enabled and trace functions are configured
- ASCDLT-846 Fixed known issue: Dlt fails to compile when DltImplementNVRamStorage is enabled but DltDevErrorDetect and DltImplementFilterMessages are disabled on AUTOSAR 4.2

- ASCDLT-849 Fixed known issue: Different ReadRamBlockFromNvmBlock function prototypes can lead to compiler warnings and/or errors

Module version 1.8.3

2020-06-19

- ASCDLT-796 Fixed known issue: Structure of the Register/UnregisterContext notification is wrong
- Implemented support for retrieving EcuId for DLT at runtime.
- ASCDLT-815 Fixed known issue: Dlt generation fails for DltVfbTracePayloadMaxSize when DltImplementVfbTrace is disabled

Module version 1.8.2

2020-02-21

- Added support for Dlt VFB tracing with arguments of the hook functions.
- ASCDLT-773 Fixed known issue: Dlt_SendTraceMessage() returns a wrong return value if Det is disabled.
- ASCDLT-780 Fixed known issue: Dlt behaves unexpectedly if message transmission is interrupted by a send log/trace request
- Added checks for valid SessionId in Dlt_SendLogMessage() and Dlt_SendTraceMessage() even if message filtering is OFF
- Changed the byte order handling of log channel names to be based on DltTupleEndianness' set value

Module version 1.8.1

2019-10-11

- ASCDLT-669 Fixed known issue: The Dlt does not take into consideration the platform's endianness when receiving AppId, ContextId and LogChannelName as control message payload
- ASCDLT-679 Fixed known issue: The GetVerboseModeStatus control message wrongly extracts the AppId/ContextId from the extended header
- ASCDLT-685 Fixed known issue: Dlt does not compile if DltGeneralStartUpDelayTimer is enabled and DltEnableTrafficShaper is disabled
- ASCDLT-686 Fixed known issue: The configured VFB traces use the wrong AppId when DltServiceAPI is set to AUTOSAR_431
- Implemented configurable behavior in case PduR_DltTransmit returns E_NOT_OK

- ASCDLT-699 Fixed known issue: GetLogInfo's LEN field is incorrect when the message is sent from the register/unregister context notification
- ASCDLT-698 Fixed known issue: Dlt could use wrong API identifiers when reporting errors to the Det.
- ASCDLT-702 Fixed known issue: When you unregister an invalid tuple, a critical section is closed before it opens
- Updated handling of AplicationId-ContextId tuples that are identical for multiple instances of the same SW-C
- ASCDLT-708 Fixed known issue: A ResetToFactoryDefault call does not erase the Dlt native block due to incorrect writing priority
- ASCDLT-704 Fixed known issue: Dlt can send unexpected log messages and/or trace messages due to wrong filtering of unregistered tuples
- ASCDLT-710 Fixed known issue: Dlt log and/or trace messages requested on satellite cores are sent on the default log channel regardless of configuration
- Implemented the GetSoftwareVersion control message.
- ASCDLT-726 Fixed known issue: Dlt uses wrong identifier for the StbM timebase when fetching the synchronized timebase

Module version 1.8.0

2019-06-14

- Implemented Logchannel concept from SWS 4.3.1
- Implemented AUTOSAR 4.3.1 APIs, control messages and operations.
- ASCDLT-569 Fixed known issue: Dlt_Rte_hook.c does not compile when trace functions are configured and DltEnableBswDistribution is enabled
- ASCDLT-560 Fixed known issue: VFB traces are not sent when there is no explicit setting of trace status via the Dlt command SetTraceStatus()
- ASCDLT-559 Fixed known issue: Trace/log messages sent by satellite cores are not filtered correctly when the log/trace settings are changed via Dlt commands using wildcards
- ASCDLT-584 Fixed known issue: Control messages that receive ApplicationId and ContextId parameters, wrongly respond with DLT_CTRL_ERROR and fail, if header endianness is LSB
- Implemented startup delay timer as specified by AUTOSAR 4.3.1.
- ASCDLT-568 Fixed known issue: SWCs are not notified when log level/trace status/verbose mode settings are changed via Dlt commands using wildcards
- ASCDLT-585 Fixed known issue: DltMaxCountContextIds is not allowed to have a value smaller than the product of DltMaxCountAppIds and DltMaxCountContextIdsPerAppId

- ASCDLT-587 Fixed known issue: Message filtering uses only the first found ContextId as input, when the same ContextId exists for multiple ApplicationIDs
- ASCDLT-615 Fixed known issue: Reception of a DLT message can fail when the message is received in multiple packages
- Updated types used for keeping information regarding the message length

Module version 1.7.1

2019-02-15

- ASCDLT-503 Fixed known issue: A store request interrupted by another store request corrupts the Dlt internal buffer

Module version 1.7.0

2018-10-26

- ASCDLT-447 Fixed known issue: Data related to context registration and log level could be out of sync on satellite cores in case of multi-core setup
- Implemented support for timestamps provided by the StbM module
- ASCDLT-452 Fixed known issue: Wrong check in control message validation can lead to out-of-bounds access
- ASCDLT-501 Fixed known issue: Dlt generates incorrectly the function pointers to RTE calls for the application notification callbacks

Module version 1.6.0

2018-06-22

- ASCDLT-351 Fixed known issue: Missing memory sections in the Dlt software component description cause uncompilable code
- Implemented verbose mode. Note: NvM layout has changed, NvM block size of NVM_BLOCK_DLT_-DATASET was increased by 1.
- Changed the data types Dlt_MessageLogInfoType and Dlt_MessageTraceInfoType to comply with the SWS. Note: External APIs (Dlt_SendLogMessage, Dlt_SendTraceMessage) are affected.
- The Dlt_RegisterContext, Dlt_SendLogMessage and Dlt_SendTraceMessage interfaces are now exposed in the new Dlt_BSW.h header file.
- ASCDLT-393 Fixed known issue: The error code assigned to DLT_E_NOT_IN_VERBOSE_MODE is wrong in Dlt_swc_interface.arxml

- Memory sections have been updated; deprecated sections have been removed.
- ASCDLT-423 Fixed known issue: Wrong Dlt instance ID is reported to Det when using multicore configuration
- ASCDLT-424 Fixed known issue: Certain control API functions that manipulate the Dlt packet header are not working if their corresponding configuration parameter is initialized to FALSE
- Implemented reception of Control Messages.

Module version 1.5.0

2018-02-23

- ASCDLT-331 Fixed known issue: Registration information is not loaded from NvM for SWCs
- Implemented compliance to MISRA-C:2012
- ASCDLT-329 Fixed known issue: Compile error occurs when Dlt is used with a non-EB NvM module
- Update according to the latest Tresos Studio

Module version 1.4.2

2017-10-09

- ASCDLT-285 Fixed known issue: Initialization process for both slave cores and master core is not correctly done
- ASCDLT-289 Fixed known issue: Buffer overflow can occur if messages from SWC are sent with appId/contextId registered from a BSWC
- ASCDLT-288 Fixed known issue: DLT reports unnecessary DET error when all datasets have been processed
- ASCDLT-316 Fixed known issue: Corruption of Dlt table data during NvM restore defaults functionality
- ASCDLT-318 Fixed known issue: Incorrect behavior for NULL values of parameters Application_ID and Context_ID in Dlt_SetLogLevel and Dlt_SetTraceStatus APIs
- Improved the usage of NvM: NvMInit callbacks made optional, NvM retry mechanism was removed
- Implemented new configuration option for configurable queue size of server ports created by the DLT assistant

Module version 1.4.1

2017-03-21

- ASCDLT-268 Fixed known issue: Restoring of the runtime data from NvM fails if a SWC already called Dlt_RegisterContext
- ASCDLT-273 Fixed known issue: Out of bounds array access happens if DltMaxCountAppIds is less than DltMaxCountContextIdsPerAppId

Module version 1.4.0

2016-12-19

- ASCDLT-215 Fixed known issue: DLT message transmission stops
- ASCDLT-223 Fixed known issue: Dlt_SendLog/TraceMessage() is not reentrant
- ASCDLT-235 Fixed known issue: Out of bound access in Dlt_AppToContextIdTable table when Dlt_RegisterContext() is called more than DltMaxCountContextIdsPerAppId times
- ASCDLT-236 Fixed known issue: DLT stops sending log Messages after restore from NVM

Module version 1.3.0

2016-11-04

- ASCDLT-206 Fixed known issue: The Message Counter (MCNT) is wrongfully reset
- ASCDLT-207 Fixed known issue: Messages in transmission might be corrupted
- ASCDLT-214 Fixed known issue: Dlt runtime variables reset to configuration values after restore from NvRAM is successful
- ASCDLT-208 Fixed known issue: Incompatible design between Dlt persistent storage functionality and NvM

Module version 1.2.2

2016-10-07

- Improved validation of the Dlt Assistant

Module version 1.2.1

2016-09-09

- ASCDLT-149 Fixed known issue: Dlt buffer indexes point to incorrect location in message buffer
- ASCDLT-144 Fixed known issue: Transmission of DLT Log messages stops after some time

- ASCDLT-156 Fixed known issue: Incompatibile code is generated due to the definition of implementation data types
- ASCDLT-150 Fixed known issue: The implementation for Rte VFB tracing functions is always generated
- ASCDLT-164 Fixed known issue: Warnings during importer run after enabling DLT BSW dsitribution
- ASCDLT-189 Fixed known issue: Det.h is always included by Dlt_Int.h

Module version 1.2.0

2016-05-25

- Implement Dlt_MainFunction()
- ASCDLT-96 Fixed known issue: Dlt_ComCopyTxData returns a wrong type if SDU length is 0
- ASCDLT-99 Fixed known issue: Dlt does not unlock the transmit buffer after it has received a negative Tx confirmation
- BSW Distribution - implementation
- Added possibility to transmit multiple Dlt frames in one Pdu

Module version 1.1.0

2015-11-25

- Implement DLT Persistency Configuration
- Implement RTE / VFB Tracing
- Implement SW-C Informer
- Implement Dlt Assistant

Module version 1.0.0

2015-07-08

- Initial version

2.2. New features

- No new features have been added since the last release.

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- New traffic shaper configuration option

  Description:

  The configuration parameter `DltEnableTrafficShaper` has been introduced in order to enable/disable the traffic shaper functionality.

  The following configuration parameters are editable if `DltEnableTrafficShaper` is enabled:

  - `DltBandwidthForComModule`
  - `DltTimePeriodTrafficShaping`
- New PDAV configuration option

  Description:

  The configuration parameter `DltCSPortsQueueLength` has been introduced in order to provide a customizable queue length for the server ports created by the Dlt.

  The value of `DltCSPortsQueueLength` controls the following behavior:

Value of `DltCSPortsQueueLength`	**Description**
0	No SERVER-COM-SPECs are created. The queue length is calculated by Rte based on the number of connected ports.
Any other value	Defines the queue size of the SERVER-COM-SPECs created by the DLT to be the configured value.

Table 2.1. Values of `DltCSPortsQueueLength`

- The Dlt_MessageLogInfoType and Dlt_MessageTraceInfoType types are now AUTOSAR compliant.
- Published interfaces for BSW usage

  Description:

  The Dlt_RegisterContext, Dlt_SendLogMessage and Dlt_SendTraceMessage interfaces are now published for BSW usage, and are located in the new Dlt_BSW.h header file, if the DltRteUsage configuration parameter is set to false.
- Different Dlt_ReturnType values based on the AUTOSAR SWS version

  Description:

Dlt_ReturnType is defined differently in the 4.2.1 and 4.2.2 AUTOSAR SWS. Based on the new configuration parameter, DltDefaultAutosarReturnValues, the customer can select with what AUTOSAR SWS (4.2.1/4.2.2) should the Dlt module comply, regarding Dlt_ReturnType's definition.

- Implemented control message service requests are also available as C-APIs

Description:

Control service request	**C-API**
Set_LogLevel	Dlt_SetLogLevel
Set_DefaultLogLevel	Dlt_SetDefaultLogLevel
Set_TraceStatus	Dlt_SetTraceStatus
Set_DefaultTraceStatus	Dlt_SetDefaultTraceStatus
Get_LogInfo, option 4	Dlt_GetLogLevel, used for one specific tuple at a time
Get_LogInfo, option 5	Dlt_GetTraceStatus, used for one specific tuple at a time
Get_DefaultLogLevel	Dlt_GetDefaultLogLevel
Get_DefaultTraceStatus	Dlt_GetDefaultTraceStatus
Store_Config	Dlt_StorePersistent
SetComInterfaceMaxBandwidth	Dlt_SetComInterfaceMaxBandwidth
GetComInterfaceMaxBandwidth	Dlt_GetComInterfaceMaxBandwidth
SetVerboseMode	Dlt_SetVerboseMode
GetVerboseModeStatus	Dlt_GetVerboseModeStatus
SetMessageFilterering	Dlt_SetMessageFiltering
GetMessageFiltereringStatus	Dlt_GetMessageFilteringStatus
SetUseECUID	Dlt_SetUseECUID
GetUseECUID	Dlt_GetUseECUID
SetUseSessionID	Dlt_SetUseSessionID
GetUseSessionID	Dlt_GetUseSessionID
UseTimestamp	Dlt_SetUseTimestamp
GetUseTimestamp	Dlt_GetUseTimestamp
UseExtendedHeader	Dlt_SetUseExtendedHeader
GetUseExtendedHeader	Dlt_GetUseExtendedHeader

Table 2.2. Control service requests and C-APIs

- Additional control APIs for global logging

  Description:

API	Description
Dlt_SetGlobalLogging	When message logging is disabled, all messages are discarded by the Dlt, regardless of log level or trace status settings
Dlt_GetGlobalLogging	Retrieves the current message logging status

Table 2.3. Additional control APIs for global logging

- The transmission of sync data between master and slave cores is done using a one to one sender-receiver channel with a configurable queue size.
- Configurable length for Application and Context description of the AppId/ContextId tuples registered in a SWC

  Description:

  Implemented support for configurable maximum length for Application and Context description information of the AppId/ContextId tuples registered in a SWC.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- No support for Dem interface

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  Interface to Dem is currently not supported.

  Rationale:

  Not planned for the current release.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00474, SWS_Dlt_00475, SWS_Dlt_00476, SWS_Dlt_00377, SWS_Dlt_-00477, SWS_Dlt_00478, SWS_Dlt_00479, SWS_Dlt_00470 AUTOSAR 4.3.1: SWS_Dlt_00474, SWS_-

Dlt_00475, SWS_Dlt_00377, SWS_Dlt_00477, SWS_Dlt_00478, SWS_Dlt_00270, SWS_Dlt_00479, SWS_Dlt_00274, SWS_Dlt_00031, SWS_Dlt_00781

- No support for Det interface

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  Interface to Det is currently not supported.

  Rationale:

  Not planned for the current release.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00430, SWS_Dlt_00431, SWS_Dlt_00376, SWS_Dlt_00480, SWS_Dlt_-00432 AUTOSAR 4.3.1: SWS_Dlt_00430, SWS_Dlt_00376, SWS_Dlt_00031, SWS_Dlt_00432

- No support for response on event

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  Response on event is currently not supported.

  Rationale:

  Not planned for the current release.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00339, SWS_Dlt_00037, SWS_Dlt_00340, SWS_Dlt_00039

- No support for ApplicationID and ContextID description

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  ApplicationID and ContextID description are not stored in NvM.

  Rationale:

ApplicationID and ContextID description are not stored when calling Dlt_RegisterContext

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00064, ECUC_Dlt_00815

- No support for SWC injection

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

Injecting data to SWC is currently not supported.

Rationale:

Not planned for the current release.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00017, SWS_Dlt_00018, SWS_Dlt_00019, SWS_Dlt_00259, SWS_Dlt_-00260, SWS_Dlt_00498, ECUC_Dlt_00819 AUTOSAR 4.3.1: SWS_Dlt_00259, SWS_Dlt_00498, SWS_-Dlt_00778, ECUC_Dlt_00847

- No support for timing messages

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

Timing messages are currently not supported.

Rationale:

Not planned for the current release.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00221, SWS_Dlt_00222

- No support DLT communication module

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The DLT module does not provide a separate communication module. The mechanism of the PduR is used instead.

Rationale:

Data transfer via UDP is provided by the AUTOSAR communication stack.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00040, SWS_Dlt_00042, SWS_Dlt_00043, SWS_Dlt_00461, SWS_Dlt_-00463, SWS_Dlt_00263, SWS_Dlt_00264, SWS_Dlt_00485, SWS_Dlt_00265

- No support for DltVfbTraceLogLevel configuration parameter

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The DltVfbTraceLogLevel configuration parameter is not supported.

Rationale:

The parameter is of no use. VFB messages are "trace" messages, not "log" messages, thus there is no "log level" which can be assigned to them.

Requirements:

AUTOSAR 4.2.1: ECUC_Dlt_00839

- DltSwcApplicationId configuration parameter is not of type string

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

DltSwcApplicationId configuration parameter is of type integer instead of type string.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00858

- DltSwcContextId configuration parameter is not of type string

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

DltSwcContextId configuration parameter is of type integer instead of type string.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00859

- DltLogChannelMaxNumOfRetries not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Implementation for DltLogChannelMaxNumOfRetries is not supported

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00884, SWS_Dlt_00761 Dlt.ASR431.Ref_SWS_Dlt_00697_NegativeTx-Confirmation_LogChannelMaxNumOfRetries

- DltLogChannelTransmitCycle not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Implementation for DltLogChannelTransmitCycle is not supported

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00885

- DltITxPduUsesTp not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Implementation for DltITxPduUsesTp is not supported

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00913

- API Gpt_GetTimeElapsed() for timestamp is not used

  Affected AUTOSAR releases:

  - R4.3 rev 1

  Description:

  No support from API Gpt_GetTimeElapsed() in getting message timestamp .

  Requirements:

  AUTOSAR 4.3.1: SWS_Dlt_00654

- The message buffer size must not be zero

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  The message buffer size must always be at least as large as DltMaxMessageLength. It is not possible to configure a message buffer size of length zero in order to force an immediate transmission of messages.

  Rationale:

  There must always be at least a small buffer in which the message and its header fields can be assembled. The DLT uses the ring buffer for this. A configuration check assures that the ring buffer is large enough for the largest possible message. When the message has been assembled, the PduR is triggered. Thus, if a "connection less interface" is connected to the PduR, the data is sent immediately and this requirement is thus implicitly fulfilled.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00493

- Scheduled function

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  The DLT module provides a scheduled function Dlt_MainFunction() which must be called periodically by the BSW scheduler.

  Rationale:

This is required for a proper implementation of the traffic shaper feature.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00468

- Different time base

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The Dlt module does not use the Gpt module for the timestamp functionality. Instead, the timestamp is provided by:

- the Os, through the usage of the Os_GetTimeStamp API (if configured to be used)
- the StbM, through the usage of the StbM_GetCurrentTime API (if configured to be used)

Rationale:

- Less configuration effort necessary
- Higher performance
- The OS provides proper functions for calculations involving the counter values, which automatically handle counter overflows

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00481 AUTOSAR 4.2.1: ECUC_Dlt_00905

- Traffic shaping

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The DLT module does not have the functionality for the diagnostic module DCM.

Requirements:

Dlt.TrafficShaping.DiagInterfaces, Dlt.TrafficShaping.DiagChannel

- Store persistent

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The Dlt module shall let the user configure the restore value for the trace status belonging to a tuple of AppId and ContextId via the callout Dlt_ApplGetConfigFactoryDefault().

Rationale:

- The information provided by the AUTOSAR SWS 2.1. regarding this parameter is not specific enough.

Requirements:

SWS_Dlt_00288, SWS_Dlt_00348

- VFB Trace message type and message info

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

In VFB tracing messages, the Message Type (MSTP) header field is set to DLT_TYPE_APP_TRACE. The Message Trace Info (MSIN) header field is set to DLT_TRACE_VFB.

Rationale:

- Requirements SWS_Dlt_00120, SWS_Dlt_00484 and SWS_00123 contradict each other. Requirements SWS_Dlt_00120 and SWS_Dlt_00123 provide the better fitting values.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00484

- Dlt_MessageTypeType

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The Dlt_MessageTypeType enum values start from 0, not from 1.

Rationale:

- Requirements SWS_Dlt_00120 and SWS_00224 contradict each other. The values mentioned in SWS_Dlt_00120 have been chosen, since enums naturally start with 0 in C.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00224

- VFB Trace Network Trace Info Type

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The Dlt_MessageNetworkTraceInfoType provides an additional DLT_NW_TRACE_MOST element.

Rationale:

- Requirements SWS_Dlt_00125 and SWS_00233 contradict each other. The additional DLT_NW_-TRACE_MOST value specified by SWS_Dlt_00125 is provided to assure compatibility and avoid compilation errors.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00233

- No support for some control messages.

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The following Control messages to DLT are currently not supported: - 0x03: Get_LogInfo, Options: 3 - 0x07: SetComInterfaceStatus - 0x0B: SetTimingPackets - 0x0C: GetLocalTime - 0x14: MessageBuffer-Overflow - 0x16: GetComInterfaceStatus - 0x17: GetComInterfaceNames - 0xFFF ... 0xFFFFFFFF: Cal-ISW-CInjection

Rationale:

Not planned for the current release.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00465, SWS_Dlt_00290, SWS_Dlt_00048, SWS_Dlt_00049, SWS_Dlt_-00050, SWS_Dlt_00051, SWS_Dlt_00201, SWS_Dlt_00501, SWS_Dlt_00197, SWS_Dlt_00502, SWS_-Dlt_00489, SWS_Dlt_00207, SWS_Dlt_00208, SWS_Dlt_00217, SWS_Dlt_00218, SWS_Dlt_00219, SWS_Dlt_00220, SWS_Dlt_00487, SWS_Dlt_00488, SWS_Dlt_00247, SWS_Dlt_00248, SWS_Dlt_-00249, SWS_Dlt_00428, Dlt.GetLogInfo.Option3.NoLogLevelNoTraceStatus, Dlt.GetLogInfo.ComInterface

- Prioritization of Dlt Control Messages responses

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  Dlt does not prioritize Control Messages responses over normal log or trace messages.

  Rationale:

  Not planned for the current release.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00490
- Accept new control messages

  Affected AUTOSAR releases:

  - R4.2 rev 1

  Description:

  Dlt does not reject new Control Messages while an old one is not finished.

  Rationale:

  Adapted due to a popular Dlt external viewer that sends Control Messages without waiting for a response for each one before sending the new requests.

  Requirements:

  AUTOSAR 4.2.1: SWS_Dlt_00417
- Several imported types are not in use

  Affected AUTOSAR releases:

  - R4.2 rev 1
  - R4.3 rev 1

  Description:

  The Dlt module does not use the following imported types:

  - Dem_DTCFormatType, Dem_EventIdType, Dem_UdsStatusByteType
  - Gpt_ChannelType, Gpt_ValueType

- NvM_BlockIdType
- StbM_SynchronizedTimeBaseType, StbM_TimeStampExtendedType

Rationale:

- The Dem types are not used because the Dem interface is not supported
- The Gpt types are not used because the Dlt uses the Os/StbM instead, for timestamp purposes
- The NvM type is not used because the Dlt does not require it
- The StbM types are not used because the Dlt does not require them

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00729

- Several optional interfaces are not in use

Affected AUTOSAR releases:

- R4.2 rev 1
- R4.3 rev 1

Description:

The Dlt module does not use the following optional interfaces:

- Dem_DltGetAllExtendedDataRecords, Dem_DltGetMostRecentFreezeFrameRecordData, Dem_GetDTCOfEvent
- Gpt_EnableNotification, Gpt_StartTimer
- StbM_GetCurrentTimeExtended

Rationale:

- The Dem optional interfaces are not used because the Dem interface is not supported
- The Gpt optional interfaces are not used because the Dlt uses the Os/StbM instead, for timestamp purposes
- The StbM optional interface is not used because the standard timestamp format is sufficient for the Dlt

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00763

- The timestamp resolution is platform-dependent

Affected AUTOSAR releases:

- R4.2 rev 1

- R4.3 rev 1

Description:

The Dlt module cannot always provide 0.1 milliseconds resolution for the timestamps.

Rationale:

The requirement implies the usage of the Gpt module for the timestamp functionality. The Gpt can provide resolutions down to 1 microseconds, hence the requirement specifies a resolution that can be provided by the Gpt. The currently implemented timestamp providers (Os/StbM) are highly platform-dependent and offer different resolutions, as a consequence.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00309 AUTOSAR 4.3.1: PRS_Dlt_00309

- Information about enabled/disabled interfaces is not stored persistently

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

When the persistent storage mechanism is enabled, the Dlt module does not persistently store any information related to the enabled or disabled state of its interfaces.

Rationale:

The functionality to enable/disable interfaces is not implemented.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00074

- Dlt command "BufferOverflowNotification" is not available

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The AUTOSAR Dlt module does not support the following Dlt Command identified by the following Services ID: ---------- | Service ID | Dlt Command Name | Description | ---------- | 0x23 | BufferOverflowNotification | Indication of a buffer overflow within the DLT module | ----------

Rationale:

The mentioned command is not currently implemented.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00643, SWS_Dlt_00670, SWS_Dlt_00776, SWS_Dlt_00777, SWS_Dlt_00760

- Dlt_GetLogInfo's "logInfo" is now a pointer to uint8

Affected AUTOSAR releases:

- R4.2 rev 1
- R4.3 rev 1

Description:

The Dlt_LogInfoType structure type used for the "logInfo" parameter of the Dlt_GetLogInfo() API is not fully compatible with the function's intended purpose.

Rationale:

The structure type only supports the storage of one AppId/ContextId tuple, which is not fully compatible with the API's intended use: the "appId" and "contextId" parameters can be given as "0", which triggers the API to write multiple entries in the structure type. Due to this issue, the "logInfo" parameter has been changed to a pointer to uint8. This can fully support multiple tuples but comes at a disadvantage because its size must be known beforehand.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00197 AUTOSAR 4.3.1: SWS_Dlt_00732

- Dlt does not respond with "DLT_NOT_SUPPORTED" when receiving deprecated commands

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

When the Dlt module receives any of the following deprecated commands, it does not respond with "DLT_NOT_SUPPORTED": * 0x07 SetComInterfaceStatus * 0x08 SetComInterfaceMaxBandwidth * 0x09 SetVerboseMode * 0x0A SetMessageFiltering * 0x0C GetLocalTime * 0x0D SetUseECUID * 0x0E SetUseSessionID * 0x0F SetUseTimestamp * 0x10 SetUseExtendedHeader * 0x14 MessageBufferOverflow * 0x16 GetComInterfacelStatus * 0x18 GetComInterfaceMaxBandwidth * 0x19 GetVerboseModeStatus * 0x1A GetMessageFilteringStatus * 0x1B GetIseECUID * 0x1C GetUseSessionID * 0x1D GetUseTimestamp * 0x1E GetUseExtendedHeader

Rationale:

The functionality is currently not implemented.

Requirements:

AUTOSAR 4.3.1: PRS_Dlt_00644

- DltGeneralRxDataPathSupport is not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The DltGeneralRxDataPathSupport configuration parameter is not used by the Dlt module to accept/reject incoming control messages. Instead, the DltRxPduId parameter is used for this purpose. If it exists and is configured correctly, the Dlt uses a macro to accept/reject incoming control messages.

Rationale:

The functionality is currently not implemented.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00698, SWS_Dlt_00699, ECUC_Dlt_00848

- DltLogChannelBufferOverflowTimer is not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

DltLogChannelBufferOverflowTimer's functionality is not supported.

Rationale:

The configuration parameter's functionality is not currently implemented.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00886, SWS_Dlt_00760

- DltLogChannelTrafficShapingBandwidth is not used

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

DltLogChannelTrafficShapingBandwidth's functionality is not supported.

Rationale:

The configuration parameter's functionality is not currently implemented.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00883, SWS_Dlt_00758, ECUC_Dlt_00849

- No support for multiple SWC instances on R4.2

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

Dlt module does not support handling of multiple SWC instances on R4.2.

Rationale:

Support for handling multiple SWC instances implies the possibility to register the same AppId/ContextId tuple for different SessionIds. This is not possible when `DltServiceAPI` configuration parameter is set to other value than AUTOSAR_431.

Requirements:

Dlt_Chap7.1.6_Implicit1

- Set/GetDefaultTraceStatus are global and not log channel specific

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The Dlt_Set/GetDefaultTraceStatus() APIs, as well as the Set/GetDefaultTraceStatus control commands/ interface operations do not change the log channel specific default trace status, but change the global default trace status instead.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00744, SWS_Dlt_00745, SWS_Dlt_00747, SWS_Dlt_00748, SWS_Dlt_-00743, SWS_Dlt_00746, SWS_Dlt_00772

- GetLogChannelNames's parameter "logChannelNames" is now of type "Dlt_LogChannelNameArrayType"

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The logChannelNames parameter of the Dlt_GetLogChannelNames() API and of the GetLogChannelNames control command and interface operation is now of type Dlt_LogChannelNameArrayType instead of Dlt_LogChannelNameType. This new array type has its size equal to the number of configured log channels.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00749, SWS_Dlt_00772

- VFB Trace Messages use 0x10000000U as Message ID

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

VFB Trace Messages currently use 0x10000000U as Message ID for the Dlt_SendTraceMessage() calls. However, SWS_Dlt_00031 states that they should use 0x00000003U for Message IDs. This functionality is not planned for the current release.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00031

- Unused DET error codes

Affected AUTOSAR releases:

- R4.2 rev 1
- R4.3 rev 1

Description:

AUTOSAR 4.2.1: Type or error | Relevance | Related error code | Value [hex] ---------- API was unable to | Development | DLT_E_INIT_FAILED | 0x07 initialize the service. | | | ---------- AUTOSAR 4.3.1: Type of error | Related error code | Value [hex] ---------- API service called with wrong parameter | DLT_E_PARAM | 0x01 Initialization failed | DLT_E_INIT_FAILED | 0x03 Registration failed | DLT_E_REGISTRATION | 0x04 ----------
The listed DET error codes are currently not used.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00447 AUTOSAR 4.3.1: SWS_Dlt_00727

- No support for runtime error reporting

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Support for runtime error code reporting is not planned for the current release.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00728

- Dlt_TriggerTransmit() is not a public API

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Dlt_TriggerTransmit() is a static function called only from Dlt_MainFunction() at the moment. Publishing of this API is not planned for the current release.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00754

- Dlt_TxFunction() is not implemented

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The Dlt_TxFunction() API is not implemented and not planned for the current release.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_91005, SWS_Dlt_00760, SWS_Dlt_00761, SWS_Dlt_00673

- DltEcuIdCalloutChoice is not configurable and DltEcuId is not used.

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The possibility to retrieve EcuIds via call-outs using DltEcuIdCalloutChoice is implemented using the current user callout approach, meaning that the name of the callout is not configurable. The Dlt_AppGetEcuIdAddress() user callout is used for EcuId retrieval.

Additionally the container DltEcuId is not used, DltEcuIdValueChoice and DltEcuIdCalloutChoice are placed in the container DltProtocol and named DltEcuId and DltEcuIdCallout respectively.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00860, ECUC_Dlt_00902, ECUC_Dlt_00862

- RxPdu configuration parameters are not implemented

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

DltIRxPduHandleId, DltIRxPduUsesTp and DltRxPduIdRef are not implemented and not planned for the current release.

Requirements:

AUTOSAR 4.3.1: ECUC_Dlt_00899, ECUC_Dlt_00900, ECUC_Dlt_00912, ECUC_Dlt_00898

- Getter functions not supported under multiple instances of SwCs

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Getter functions for log level and trace status do not currently support multiple instances of SwCs, as AR specifications do not provide argument for session IDs, in order to distinguish identical tuples under different sessions.

Requirements:

Dlt.ASR431.Chap7.1.2_Implicit1

- Several GetSoftwareVersion requirements are not implemented

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The requirements listed below are for the external client and cannot be implemented by the Dlt module.

Rationale:

The Dlt module cannot detect when a conection is established. The Dlt is initialized and then waits until the startup delay timer has elapsed. The messages are sent afterwards until the buffers are empty. The external client is responsible for making sure that the first control message sent is GetSoftwareVersion.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00394, SWS_Dlt_00395, SWS_Dlt_00492

- No support for post-build selectable behavior

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The Dlt module does not support post-build behavior for any parameter.

Requirements:

ECUC_Dlt_00800, ECUC_Dlt_00831, ECUC_Dlt_00802, ECUC_Dlt_00803, ECUC_Dlt_00835, ECUC_Dlt_00805, ECUC_Dlt_00843, ECUC_Dlt_00807, ECUC_Dlt_00806, ECUC_Dlt_00811, ECUC_Dlt_00812, ECUC_Dlt_00813, ECUC_Dlt_00814, ECUC_Dlt_00836, Dlt.ASR431.ECUC_Dlt_00870, Dlt.ASR431.ECUC_Dlt_00871, Dlt.ASR431.ECUC_Dlt_00895, Dlt.ASR431.ECUC_Dlt_00874, Dlt.ASR431.ECUC_Dlt_00874, Dlt.ASR431.ECUC_Dlt_00887, Dlt.ASR431.ECUC_Dlt_00896, Dlt.ASR431.ECUC_Dlt_00888, Dlt.ASR431.ECUC_Dlt_00864, Dlt.ASR431.ECUC_Dlt_00889, Dlt.ASR431.ECUC_Dlt_00886, Dlt.ASR431.ECUC_Dlt_00877, Dlt.ASR431.ECUC_Dlt_00884, Dlt.ASR431.ECUC_Dlt_00878, Dlt.ASR431.ECUC_Dlt_00883, Dlt.ASR431.ECUC_Dlt_00879, Dlt.ASR431.ECUC_Dlt_00893, Dlt.ASR431.ECUC_Dlt_00892, Dlt.ASR431.ECUC_Dlt_00913, Dlt.ASR431.ECUC_Dlt_00856, Dlt.ASR431.ECUC_Dlt_00854, Dlt.ASR431.ECUC_Dlt_00858, Dlt.ASR431.ECUC_Dlt_00859, Dlt.ASR431.ECUC_Dlt_00812, Dlt.ASR431.ECUC_Dlt_00911

- Message counter is increased individually per log channel

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

The message counter in the MCNT field of each Dlt message is currently incremented per log channel instead of globally.

Rationale:

The PRS_Dlt_00105 requirement specifies that a generic message counter shall be implemented, which shall count every log and trace message received via the Dlt API. This somewhat contradicts SWS_Dlt_-00671, which states that a message counter shall be implemented for each specific log channel. Due to this conflict, both requirements cannot be implemented. As such, the behavior stated in SWS_Dlt_00671 has been chosen for implementation.

Requirements:

AUTOSAR 4.3.1: PRS_Dlt_00105

- Size of internal Dlt buffer for message processing is set via a single configuration parameter

Affected AUTOSAR releases:

- R4.2 rev 1

Description:

The size of the internal Dlt buffer used for log and trace messages is determined by a specific chosen parameter.

Rationale:

The SWS_Dlt_00003 requirement specifies that the size of the internal Dlt buffer, used prior to the initialization of the module (and re-used later as normal buffer if desired), shall be determined by the configuration parameter DltInitBufferSize (user definable). The specifications however mention other buffers as well, each with its size determined by a different configuration parameter (for example DltMessageBufferSize, from requirement SWS_Dlt_00342). Because of this, for the sake of memory efficiency a single internal buffer has been implemented for general usage (pre-initialization, runtime, no external client connected), with its size determined by the parameter DltMessageBufferSize, as required by SWS_Dlt_00342. This leaves the DltInitBufferSize parameter unused, thus deviating from the SWS_Dlt_00003 requirement.

Requirements:

AUTOSAR 4.2.1: SWS_Dlt_00003

- Dlt will filter messages with a LogLevel higher than the configured value of the LogChannel threshold or the Log Level of the found ApplicationID/ContextId

Affected AUTOSAR releases:

- R4.3 rev 1

Description:

Log messages with a LogLevel higher than the configured value of log level threshold of the element for which the Dlt_SendLogMessage() API was called. Log messages with a LogLevel higher than the configured value of LogChannel threshold for the identified LogChannel shall be discarded and E_OK shall be returned. This shall be done on each LogChannel from the list of output LogChannels for the LogMessages.

Rationale:

These requirements are wrong (AUTOSAR ticket AR-96503), the logic for filtering messages was reversed when these requirements were introduced with AUTOSAR 4.3.0 SWS. The correct behavior is the one presented in AUTOSAR 4.2.2 SWS: Dlt shall check if the log level of the incoming log message is the same or below as the maximum log level stored for this Context ID - Application ID tuple (the log level of the incoming message shall be in the pass through range). If the check is not successful, the messages shall be discarded, otherwise the message shall be transmitted to the external client.

Requirements:

AUTOSAR 4.3.1: SWS_Dlt_00662, SWS_Dlt_00667

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- Limitation of number of session IDs

  Description:

  The number of session IDs which can be registered by SWCs is limited to 2147483647.

  Rationale:

  This is the maximum value supported by the `size()` method of Java's `Lists<>` class, which is needed for evaluating the system description files.

  Requirements:

  SWS_Dlt_225

- AppId/ContextId registration cannot differ only by SessionId

  Description:

When DltServiceAPI is set to "AUTOSAR_421" or "AUTOSAR_422", only one SessionId (port belonging to a SWC) can be registered per tuple of AppId/ContextId.

Rationale:

The current internal data structure does not allow to have the same AppId/ContextId mapped to different SessionIds to have more optimal data handling.

- Limitation of traffic shaping

  Description:

  The minimum limit for traffic shaping is 1kByte/sec.

  Rationale:

  The configuration parameter DltBandwidthForComModule expects the value in kbit/sec, but the limiting of traffic can only be done on whole bytes.

- Limitation of maximum number of contexts and maximum message length

  Description:

  When DltEnableBswDistribution is set to true, the maximum number of possible contexts registered is limited to 4095 and the maximum message length is limited to 8188. These values depend on the architecture on which the stack is configured.

  Rationale:

  When multi-core is enabled, data needs to be exchanged through the IOC module between the satellites and the master to ensure a consistent configuration. The size of the data exchanged through the IOC cannot exceed 65536. To meet this constraint, when BSW Distribution is enabled, maximum values were computed for DltMaxCountContextIds and DltMaxMessageLength. If these parameters are configured with a greater value, the project was risked not to compile as the compilation is stopped in the IOC code.

- Limitation related to the basic software distribution functionality

  Description:

  When DltEnableBswDistribution is set to true, control APIs can only be called from the master context in order to modify the status of the following run-time variables: DltDefaultTraceStatus, DltFilterMessages, DltDefaultMaxLogLevel, DltHeaderUseEcuId, DltHeaderUseExtendedHeader, DltHeaderUseSessionID, DltHeaderUseTimestamp. If these functions are called from the slave context, a negative return value is reported because these APIs cannot be called from the satellite cores.

  Rationale:

Run-time variables are only stored in the master core. Therefore, the necessity to only call the control APIs from the master core is justified.

Description:

If DltImplementFilterMessages and DltFilterMessages are enabled, then DLT filtering functionality is available for both master core and satellite cores. By using the filtering functionality on the slave core, a lot of inter-core communication can be avoided, because each message which does not pass the filter can simply be discarded by the slave and does not need to be forwarded to the master.

Description:

If DltImplementNVRamStorage is enabled, then the persistent storage functionality shall be possible from both master core and slave core. Note that on the satellite core, the slave sends the session ID to the master Dlt instance via SchM send call. On the master core this then triggers a function which receives the session ID of a session from the slave and triggers the Dlt_StorePersistent() API.

Description:

If the satellite core calls Dlt_SendLogMessage()/ Dlt_SendTraceStatus() but the processing of the log message/trace status fails on the master's side, the slave will never be informed about this.

Rationale:

If a Dlt_SendLogMessage()/Dlt_SendTraceStatus() request fails on the master core with a negative return value, the information is not passed back from the master to the slave core.

- Limitation functionality related to some input parameters

Description:

Dlt_RegisterContext() API will not use the input parameters app_description, len_app_description, context_description, len_context_description.

Rationale:

To reduce the size of memory footprint, app_description and context_description is ignored in the current implementation.

Requirements:

SWS_Dlt_00245, ECUC_Dlt_00836

- Reception PDU Ids

Description:

Only one Rx PDU Id can be used for receiving messages.

Rationale:

A maximum multiplicity of 1 for DltRxPdu should be sufficient for most use cases.

- Mapping of error codes for Dlt_SendLogMessage API.

Description:

For Autosar 4.1 and 4.2: If Det is enabled and Dlt_SendLogMessage() is called with an options parameter different from DLT_TYPE_LOG in the log_info structure, the function shall report a DLT_E_WRONG_PARAMETERS error to Det and exit with return value DLT_E_ERROR_UNKNOWN. If Det is enabled and Dlt_SendLogMessage() is called with a structure containing the relevant information for filtering the message (log_info) or the payload of the send message (log_data) as NULL_PTR, the function shall report a DLT_E_PARAM_POINTER error to Det and exit with return value DLT_E_ERROR_UNKNOWN. For Autosar 4.3: If Det is enabled and Dlt_SendLogMessage() is called with an options parameter different from DLT_TYPE_LOG in the log_info structure, the function shall report a DLT_E_WRONG_PARAMETERS error to Det and exit with return value DLT_E_UNKNOWN_SESSION_ID. If Det is enabled and Dlt_SendLogMessage() is called with a structure containing the relevant information for filtering the message (log_info) or the payload of the send message (log_data) as NULL_PTR, the function shall report a DLT_E_PARAM_POINTER error to Det and exit with return value DLT_E_UNKNOWN_SESSION_ID.

Rationale:

In Autosar 4.3 DLT_E_ERROR_UNKNOWN is no longer a valid return value for Dlt_SendLogMessage. DLT_E_UNKNOWN_SESSION_ID is the semantically closest out of all the available return values.

- Mapping of error codes for Dlt_SendTraceMessage API.

Description:

For Autosar 4.1 and 4.2: If Det is enabled and Dlt_SendTraceMessage() is called with an options parameter different from DLT_TYPE_APP_TRACE and DLT_TYPE_NW_TRACE in the trace_info structure, the function shall report a DLT_E_WRONG_PARAMETERS error to Det and exit with return value DLT_E_ERROR_UNKNOWN. If Det is enabled and Dlt_SendTraceMessage() is called with a structure containing the relevant information for filtering the message (trace_info) or the payload of the send message (trace_data) as NULL_PTR, the function shall report a DLT_E_PARAM_POINTER error to Det and exit with return value DLT_E_ERROR_UNKNOWN. For Autosar 4.3: If Det is enabled and Dlt_SendTraceMessage() is called with an options parameter different from DLT_TYPE_APP_TRACE and DLT_TYPE_NW_TRACE in the trace_info structure, the function shall report a DLT_E_WRONG_PARAMETERS error to Det and exit with return value DLT_E_UNKNOWN_SESSION_ID. If Det is enabled and Dlt_SendTraceMessage() is called with a structure containing the relevant information for filtering the message (trace_info) or the payload of the send message (trace_data) as NULL_PTR, the function shall report a DLT_E_PARAM_POINTER error to Det and exit with return value DLT_E_UNKNOWN_SESSION_ID.

Rationale:

In Autosar 4.3 DLT_E_ERROR_UNKNOWN is no longer a valid return code. DLT_E_UNKNOWN_SESSION_ID is the semantically closest out of all the available return values.

- Limitation of setter API functions, control messages and operations

  Description:

  If a setter function is called with a valid tuple, all identical tuples under all SWCs will be updated with the requested change.

  Rationale:

  The Autosar APIs Dlt_SetLogLevel(), Dlt_SetTraceStatus(), and Dlt_SetVerboseMode() control messages (from the DltControlService interface), do not provide a way to distinguish between identical tuples under different SWCs. This makes setting a specific tuple under a particular SWC impossible, in scenarios where there are multiple identical tuples under different SWCs. Because of this, the setter function APIs requested for a specific tuple will affect all identical tuples.

2.6. Open-source software

`Dlt` does not use open-source software.