

Document Title	SWS_DiagnosticLogAndTrace: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SWS_DiagnosticLogAndTrace	3
1.1	Specification Item ECUC_Dlt_00809	3
1.2	Specification Item ECUC_Dlt_00812	6
1.3	Specification Item ECUC_Dlt_00832	9
1.4	Specification Item ECUC_Dlt_00900	12
1.5	Specification Item ECUC_Dlt_00905	15
1.6	Specification Item ECUC_Dlt_00907	18
1.7	Specification Item ECUC_Dlt_00911	21
1.8	Specification Item ECUC_Dlt_00912	24
1.9	Specification Item ECUC_Dlt_00913	27
1.10	Specification Item ECUC_Dlt_00914	30
1.11	Specification Item SWS_Dlt_00270	33
1.12	Specification Item SWS_Dlt_00272	34
1.13	Specification Item SWS_Dlt_00273	36
1.14	Specification Item SWS_Dlt_00274	38
1.15	Specification Item SWS_Dlt_00482	39
1.16	Specification Item SWS_Dlt_00515	40
1.17	Specification Item SWS_Dlt_00516	42
1.18	Specification Item SWS_Dlt_00654	55
1.19	Specification Item SWS_Dlt_00655	57
1.20	Specification Item SWS_Dlt_00692	60
1.21	Specification Item SWS_Dlt_00729	63
1.22	Specification Item SWS_Dlt_00754	67
1.23	Specification Item SWS_Dlt_00756	68
1.24	Specification Item SWS_Dlt_00763	70
1.25	Specification Item SWS_Dlt_00774	75
1.26	Specification Item SWS_Dlt_91006	78
1.27	Specification Item SWS_Dlt_91007	80
1.28	Specification Item SWS_Dlt_91008	82

1 SWS_DiagnosticLogAndTrace

1.1 Specification Item ECUC_Dlt_00809

Trace References:

none

Content:

Container Name	DltGeneralDltGeneral
Description	This container lists all the global Dlt functionalities that can be enabled or disabled at pre-compile time to optimize resource consumption.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
DltGeneralDevErrorDetect	ECUC_Dlt_00840
DltGeneralInjectionSupport	ECUC_Dlt_00847
DltGeneralRegisterContextNotification	ECUC_Dlt_00846
DltGeneralRxDataPathSupport	ECUC_Dlt_00848
DltGeneralStartUpDelayTimer	ECUC_Dlt_00897
DltGeneralTimeStampSupport	ECUC_Dlt_00850
DltGeneralTrafficShapingSupport	ECUC_Dlt_00849
DltGeneralVersionInfoApi	ECUC_Dlt_00844
DltGeneralGptChannelRef	ECUC_Dlt_00905
DltGeneralNvRamRef	ECUC_Dlt_00845
DltGeneralStbMTimeBaseRef	ECUC_Dlt_00914

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT

Document: AUTOSAR_SWS_DiagnosticLogAndTrace

Chapter: All

Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt

module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.2 Specification Item ECUC_Dlt_00812

Trace References:

none

Content:

Name	DltUseExtHeaderInNonVerbModeDltProtocol.DltUseExtHeaderInNonVerbMode		
Parent Container	DltProtocol		
Description	Non Verbose messages (opposed to verbose messages) do not need an extended header. If this flag is set to true the extended header shall also be used for non verbose messages. If DltGeneralNvRAMSupport is enabled this parameter is the initial value for the corresponding NVRam entry. If DltGeneralNvRAMSupport is not set, Link-Time or Post-Build configuration shall be used.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: Can only be true if DltImplementExtendedHeader is true.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace

Chapter: All

Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
- Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
- Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
- SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
- No prototype is defined for the EcuID call-out function (DltEcuIdCalloutChoice).
- According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
- Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
- Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
- Chapter 7 contains many references to non-existent parameters.
- Clarification is required regarding MCNT.

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized

time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.3 Specification Item ECUC_Dlt_00832

Trace References:

none

Content:

Container Name	DltProtocolDltProtocol
Description	Configuration parameters for handling the specific protocol variants.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
DltHeaderUseEculd	ECUC_Dlt_00811
DltHeaderUseSessionID	ECUC_Dlt_00813
DltHeaderUseTimestamp	ECUC_Dlt_00814
DltUseExtHeaderInNonVerbMode	ECUC_Dlt_00812
DltUseVerboseMode	ECUC_Dlt_00911

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DltEculd	1	This is a choice container to choose between a Eculd value or a callout to get the Eculd.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder

Role: Reviewer

Finding in SWS DLT

Document: AUTOSAR_SWS_DiagnosticLogAndTrace

Chapter: All

Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFuncton () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.4 Specification Item ECUC_Dlt_00900

Trace References:

none

Content:

Container Name	DltRxPduDltRxPdu		
Description	Contains the Pdu IDs to be used for Dlt control messages reception.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
DltRxPduHandleId	ECUC_Dlt_00899
DltRxPduUsesTp	ECUC_Dlt_00912
DltRxPduIdRef	ECUC_Dlt_00898

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIdCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:
Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1
Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set

to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.5 Specification Item ECUC_Dlt_00905

Trace References:

none

Content:

Name	DltGeneralGptChannelRefDltGeneral.DltGeneralGptChannelRef		
Parent Container	DltGeneral		
Description	If TimeStampSupport is used the Dlt module shall fetch the time from the Gpt module by calling Gpt_GetTimeElapsed with the here referenced GptChannel. The tick duration can be deduced from the GptChannelTickFrequency parameter of the GptChannelConfiguration container. This is necessary to calculate the microsecond resolution timestamp output in the Dlt message. A GPT timer shall be used which starts with value 0 at ECU start-up according to the PRS Dlt Protocol Specification.		
Multiplicity	0..1		
Type	Symbolic name reference to [GptChannelConfiguration]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: DltGeneralTimeStampSupport is set to TRUE and DltGeneralStbMTimeBaseRef is not configured.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of AppID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIdCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:
Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.6 Specification Item ECUC_Dlt_00907

Trace References:

none

Content:

Container Name	DltTxPduDltTxPdu
Description	Contains the configuration parameters of the AUTOSAR Dlt module's Tx Pdus.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
DltTxPduHandleId	ECUC_Dlt_00893
DltTxPduUsesTp	ECUC_Dlt_00913
DltTxPduIdRef	ECUC_Dlt_00892

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of AppID and ContextID.
 - Dlt_TxFUNCTION () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:
Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.7 Specification Item ECUC_Dlt_00911

Trace References:

none

Content:

Name	DltUseVerboseModeDltProtocol.DltUseVerboseMode		
Parent Container	DltProtocol		
Description	If this flag is set to TRUE, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIdCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:
Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1
Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set

to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.8 Specification Item ECUC_Dlt_00912

Trace References:

none

Content:

Name	DltIRxPduUsesTpDltRxPdu.DltIRxPduUsesTp		
Parent Container	DltRxPdu		
Description	If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFUNCTION () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIdCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the

API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneral-GptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUse-Timestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

—Last change on issue 76283 comment 15—

BW-C-Level:

Application	Specification	Bus
1	3	4

1.9 Specification Item ECUC_Dlt_00913

Trace References:

none

Content:

Name	DltTxPduUsesTpDltTxPdu.DltTxPduUsesTp		
Parent Container	DltTxPdu		
Description	If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	—		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.10 Specification Item ECUC_Dlt_00914

Trace References:

none

Content:

Name	DltGeneralStbMTimeBaseRefDltGeneral.DltGeneralStbMTimeBaseRef		
Parent Container	DltGeneral		
Description	If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase.		
Multiplicity	0..1		
Type	Symbolic name reference to [StbMSynchronizedTimeBase]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:
Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set

to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.11 Specification Item SWS_Dlt_00270

Trace References:

SRS_Dlt_00007

Content:

Only if Dem_DltGetAllExtendedDataRecords returns with E_OK, the provided information shall be further processed by the DLT module. No retry nor error handling shall be done.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76472: [WORD] duplicate sws artifacts in word documents

Problem description:

Name: Technical Office
Phone:
Role:

Description/Motivation:

The artifact analysis checker found some documents where sws artifact duplication occurs (same tables). These are just minor copy paste errors.

Agreed solution:

Please check the included artifact/table and then replace the table with the correct one.

The generated artifacts can be found here:

https://svn.autosar.org/repos/work/26_Standards/10_CP_R4/01_Sources/ZGEN_SWSArtifacts/ch

https://svn.autosar.org/repos/work/26_Standards/10_CP_R4/01_Sources/ZGEN_SWSArtifacts/S

Please search in the htmls for the artifact to be placed, copy the anchorname and place it in the word file via SHIFT+F9

SWS_CryptoServiceManager Table 7.11.2.41 Csm_SymBlockEncryptResultBuffer /Service_Interfaces/HTML/Csm.html SymBlockEncryptDataBuffer ERROR is included at least twice

SWS_CryptoServiceManager Table 7.11.2.63 Csm_SignatureGenerateResultType_Crypto /Service_Interfaces/HTML/Csm.html Csm_SignatureGenerateDataType ERROR is included at least twice

SWS_DiagnosticLogAndTrace Table 8.4.6 Dlt_DemTriggerOnEventData /chap08/Dlt.html Dlt_DemTriggerOnEventData is included at least twice

SWS_DiagnosticLogAndTrace Table 8.12 Dlt_MessageLogInfoType /Service_Interfaces/HTML/Dlt.html Dlt_MessageLogInfoType is included at least twice

SWS_EthernetInterface Table 8.6.3 Configurable interfaces /chap08/EthIf.html EthIf_SwitchEgressTimeStampIndication is included at least twice

SWS_EthernetInterface Table 8.6.3 Configurable interfaces /chap08/EthIf.html EthIf_SwitchIngressTimeStampIndication is included at least twice

–Last change on issue 76472 comment 13–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item SWS_Dlt_00272

Trace References:

none

Content:

Service name:	Dlt_RxIndicationDlt_RxIndication	
Syntax:	<pre>void Dlt_RxIndication(PduIdType RxPduId, const PduInfoType* PduInfoPtr)</pre>	
Service ID[hex]:	0x42	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	RxPduIdDlt_RxIndication.RxPduId	ID of the received PDU.
	PduInfoPtrDlt_RxIndication.PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Indication of a received PDU from a lower layer communication interface module.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception,

Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.13 Specification Item SWS_Dlt_00273

Trace References:

none

Content:

Service name:	Dlt_TxConfirmationDlt_TxConfirmation	
Syntax:	void Dlt_TxConfirmation(PduldType TxPduld, Std_ReturnType result)	
Service ID[hex]:	0x40	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
Parameters (in):	TxPduldDlt_TxConfirmation.TxPduld	ID of the PDU that has been transmitted.
	resultDlt_TxConfirmation.result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.14 Specification Item SWS_Dlt_00274

Trace References:

SRS_Dlt_00007

Content:

Only if Dem_DltGetMostRecentFreezeFrameRecordData returns with E_OK, the provided information shall be further processed by the DLT module. No retry nor error handling shall be done.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76472: [WORD] duplicate sws artifacts in word documents

Problem description:

Name: Technical Office

Phone:

Role:

Description/Motivation:

The artifact analysis checker found some documents where sws artifact duplication occurs (same tables). These are just minor copy paste errors.

Agreed solution:

Please check the included artifact/table and then replace the table with the correct one.

The generated artifacts can be found here:

https://svn.autosar.org/repos/work/26_Standards/10_CP_R4/01_Sources/ZGEN_SWSArtifacts/cl

https://svn.autosar.org/repos/work/26_Standards/10_CP_R4/01_Sources/ZGEN_SWSArtifacts/S

Please search in the htmls for the artifact to be placed, copy the anchorname and place it in the word file via SHIFT+F9

SWS_CryptoServiceManager Table 7.11.2.41 Csm_SymBlockEncryptResultBuffer /Service_Interfaces/HTML/Csm.html SymBlockEncryptDataBuffer ERROR is included at least twice

SWS_CryptoServiceManager Table 7.11.2.63 Csm_SignatureGenerateResultType_Crypto /Service_Interfaces/HTML/Csm.html Csm_SignatureGenerateDataType ERROR is included at least twice

SWS_DiagnosticLogAndTrace Table 8.4.6 Dlt_DemTriggerOnEventData /chap08/Dlt.html Dlt_DemTriggerOnEventData is included at least twice

SWS_DiagnosticLogAndTrace Table 8.12 Dlt_MessageLogInfoType /Service_Interfaces/HTML/Dlt.html Dlt_MessageLogInfoType is included at least twice

SWS_EthernetInterface Table 8.6.3 Configurable interfaces /chap08/EthIf.html EthIf_SwitchEgressTimeStampIndication is included at least twice

SWS_EthernetInterface Table 8.6.3 Configurable interfaces /chap08/EthIf.html EthIf_SwitchIngressTimeStampIndication is included at least twice

–Last change on issue 76472 comment 13–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.15 Specification Item SWS_Dlt_00482

Trace References:

none

Content:

The module header file Dlt.h shall include Rte_Dlt_Type.h to include the types, which are commonly used by BSW Modules and Software Components. Dlt.h and all Dlt*cfg.h files shall only contain types, that are not already defined in Rte_Dlt_Type.h.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74862: [SRS DLT] Remove the mandatory dependence to Rte from Dlt

Problem description:

While looking at the Dlt requirements I noticed the strong mandatory dependency to the Rte.

For instance, SWS_Dlt_00482 states that

"[...] The module header file Dlt.h shall include Rte_Dlt_Type.h [...]"

Diagram "Figure 4 Header file structure recommended for Dlt source" depicts this dependency.

However, I see no reason why the Rte interface should be mandatory.

According to the definition of the module in SWS_Dlt_00464:

"Dlt is a basic software module which handles and stores log and trace messages produced by SW-C it self or the interactions between SW-C and RTE/VFB and by the Basic Software Modules Dem and Det."

Therefore, there is an use-case (Dem and Det) where the Rte isn't necessary at all.

It makes no sense to enforce the Rte dependency just because there are 2 other use-cases (SWC's or VFB tracing) that require it.

I would propose to introduce a new general boolean parameter, DltRteUsage and only perform the Rte_Dlt inclusions if this parameter is enabled.

Agreed solution:

Remove SWS_Dlt_00482

–Last change on issue 74862 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item SWS_Dlt_00515

Trace References:

SRS_Dlt_00034

Content:

Service name:	DltCom_CopyRxDataDltCom_CopyRxData
Syntax:	BufReq_ReturnType DltCom_CopyRxData(PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr)
Service ID[hex]:	0x44

Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDltCom_CopyRxData.id	Identification of the received I-PDU.
	infoDltCom_CopyRxData.info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout):	None	
Parameters (out):	bufferSizePtrDltCom_CopyRxData.bufferSizePtr	Available receive buffer after data has been copied.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description:	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining data buffer is written to the position indicated by bufferSizePtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77935: [PduR] Misleading description of CopyRxData

Problem description:

Name: Martin Schlodder
Role: Member of WP-A2

Description/Motivation:

The description of the CopyRxData API says: "The size of the remaining data is written to the position indicated by bufferSizePtr."

This text seems to have been copied from the CopyTxData call, where it is correct. CopyRxData should talk about "remaining buffer", not "remaining data".

Agreed solution:

In the description of the API PduR_<User:LoTp>CopyRxData (SWS_PduR_00512), replace "remaining data" by "remaining buffer".

BW-C-Level:

Application	Specification	Bus
1	1	1

1.17 Specification Item SWS_Dlt_00516

Trace References:

SRS_Dlt_00034

Content:

Service name:	DltCom_CopyTxDataDltCom_CopyTxData	
Syntax:	BufReq_ReturnType DltCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)	
Service ID[hex]:	0x43	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDltCom_CopyTxData.id	Identification of the transmitted I-PDU.
	infoDltCom_CopyTxData.info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retryDltCom_CopyTxData.retry	This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems. If the retry parameter is a NULL_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element. If TpDataState indicates TP_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the current data copy position.
Parameters (inout):	None	

Parameters (out):	availableDataPtrDltCom_CopyTx Data.availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrlsoTp) to determine the size of the following CFs.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied. BUFREQ_E_NOT_OK: Data has not been copied. Request failed.
Description:	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->Tp DataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68035: [diverse] Introduce rules defining which input parameters shall be passed per value and which ones per const reference

Problem description:

SWS_BSW_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

In addition to that there shall be a SWS item that states that input parameters of integral and enum type shall be passed by value whereas input parameters of structure type shall be passed by reference.

The various transformer SWS documents shall be adapted accordingly.

—Last change on issue 68035 comment 4—

Agreed solution:

BSW UML model

The attachment "Changed Proposal in WP-A meeting" contains a list of changes to the APIs in the model (see column H). Afterwards all related documents (included in impact list) shall update their generated artifacts.

General Requirements on Basic Software Modules

~~~~~

Introduce the following requirements prior to SRS\_BSW\_00371:

SRS\_BSW\_xxxxx: Input parameters of scalar and enum types shall be passed as a value.

Type: valid

Description: All input parameters of scalar or enum type shall be passed as a value.

Rationale:

Use case: For example a function named <Mip>\_SomeFunction with a return type of Std\_ReturnType and a single parameter named SomeParameter of type uint8 is defined with the following signature:

Std\_ReturnType <Mip>\_SomeFunction(uint8 SomeParameter);

Dependencies: —

Supporting Material: —

SRS\_BSW\_yyyyy: Input parameters of structure type shall be passed as a reference to a constant structure

Type: valid

Description: All input parameters of structure type shall be passed as a reference constant structure

Rationale: Passing input parameters of structure type by value would result in additional run-time overhead due to efforts for copying the whole structure.

Use case: For example a function named <Mip>\_SomeFunction with a return type of Std\_ReturnType and a single parameter named SomeParameter of type SomeStructure (where SomeStructure is a struct) is defined with the following signature:

Std\_ReturnType <Mip>\_SomeFunction(P2CONST(SomeStructure, AUTOMATIC, <MIP>\_APPL\_DATA) SomeParameter);

Dependencies: —

Supporting Material: —

SRS\_BSW\_zzzzz: Input parameters of array type shall be passed as a reference to the constant array base type

Type: valid

Description: All input parameters of array type shall be passed as a reference to the constant array base type

Rationale: This effectively matches the behavior specified in the ISO-C:90 namely that a "declaration of a parameter as 'array of type' shall be adjusted to 'qualified pointer to type'".

Use case: For example a function named <Mip>\_SomeFunction with a return type of Std\_ReturnType and a single parameter named SomeParameter of type array of uint8 is defined with the following signature:

```
Std_ReturnType      <Mip>_SomeFunction(P2CONST(uint8,      AUTOMATIC,
<MIP>_APPL_DATA) SomeParameter);
```

Dependencies: –

Supporting Material: —

## General Specification of Transformers

~~~~~

In SWS_Xfrm_00036 change

```
const <type>* dataElement
```

to

```
<paramtype> dataElement
```

and add the following to the where clause after the API table after the bullet "type is data type of the data element"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00038 change

```
[<type> data_1,] ...
```

```
[<type> data_n]
```

to

```
[<paramtype> data_1,] ...
```

[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy,
and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the
transformer as data_1, ..., data_n the requirements to API parameters stated in
chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017],
[SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_Xfrm_00040 change

[<originalData1>, ...
<originalDataN>]

to

[<paramtype> originalData1,] ...
[<paramtype> originalDataN]

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy,
and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

In SWS_Xfrm_00044 change

<type> *data_1, ...

<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_XXXXX, SRS_BSW_YYYYY,
and SRS_BSW_ZZZZZ) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the
transformer as data_1, ..., data_n the requirements to API parameters stated in
chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017],
[SWS_Rte_01018] and [SWS_Rte_05107]).

Speci?cation of SOME/IP Transformer

~~~~~

In SWS\_SomeIpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY,  
and SRS\_BSW\_ZZZZZ) and SWS BSW General (see SWS\_BSW\_00186 and

SWS\_BSW\_00187).

In SWS\_SomelpXf\_00141 change

[<type> data\_1,] ...  
[<type> data\_n]

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy,  
and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the  
transformer as data\_1, ..., data\_n the requirements to API parameters stated in  
chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017],  
[SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

In SWS\_SomelpXf\_00145 change

<type> \*data\_1, ...  
<type> \*data\_n

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element



"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy, and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data\_1, ..., data\_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

Specification of COM Based Transformer

~~~~~

In SWS_ComXf_00007 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet "type is data type of the data element

"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

Specification of Time Sync over Ethernet

~~~~~

In SWS\_EthTSyn\_00040 make the parameter DataPtr of EthTSyn\_RxIndication const.

## Specification of SWS FlexRay Interface

~~~~~

Change SWS_Frlf_05073 from
Frlf_NumOfStartupFramesPtr (IN)
to
Frlf_NumOfStartupFramesPtr (OUT)

Specification of ADC

~~~~~

~[SWS\_Adc\_00419] Adc\_SetupResultBuffer: change Adc\_ValueGroupType\* to  
const Adc\_ValueGroupType\*  
~[SWS\_Adc\_00369] Adc\_ReadGroup: move Adc\_ValueGroupType \* from Parame-  
ters (in) to Parameters (out)

There is no need to change parameter from IN to INOUT in Adc\_SetupResultBuffer

## Specification of Com

~~~~~

Change type of parameter MetaData of Com_TriggerIPDUSendWithMetaData from
uint8* to const uint8*

Specification of ComM

~~~~~

no change required

## Specification of Dem

~~~~~

no change required

Specification of DLT

~~~~~

no change required

## Specification of DoIP

~~~~~

From:

Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed,
uint8* ConfirmationReqData, uint8* ConfirmationResData)

Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenti-
fied, uint8* AuthenticationReqData, uint8* AuthenticationResData)

To:

Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed,
const uint8* ConfirmationReqData, uint8* ConfirmationResData)

Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenti-
fied, const uint8* AuthenticationReqData, uint8* AuthenticationResData)

Specification of E2ELibrary

~~~~~

no change required

## Specification of Eth

~~~~~

no change required

Specification of EthIf

~~~~~

no change required

## Specification of EthSwitchDriver

~~~~~

no change required

Specification of ICUDriver

~~~~~

SWS\_Icu\_00201: Icu\_StartTimestamp

Parameter (IN): Icu\_ValueType\* BufferPtr shall be changed to Parameters (out) type

## Specification of LdCom

~~~~~

[SWS_LDCOM_00027]: LdCom_CopyTxData
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

[SWS_LDCOM_00036]: Rte_LdComCbKCopyTxData_<sn>
BufReq_ReturnType Rte_LdComCbKCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType Rte_LdComCbKCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin

~~~~~

PduInfoPtr needs to be const in Std\_ReturnType Lin\_SendFrame( uint8 Channel, const Lin\_PduType\* PduInfoPtr )

#### Specification of PduR

~~~~~

* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm

~~~~~

Change parameter 'name' of User\_AddressClaimedIndication to type 'const uint8'

#### Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver

~~~~~

=> nothing to change for SWS SPI

#### Specification of SynchronizedTimeBaseManager

~~~~~

"StbM not affected. All issues listed in the WP-A attachment have been already

implemented by IT 69124 in context of RfC 65633"

Specification of Tcplp

~~~~~

~[SWS\_TCPIP\_00040] Tcplp\_DhcpReadOption: change DataPtr from (IN) to (OUT)

~[SWS\_TCPIP\_00189] Tcplp\_DhcpV6ReadOption: change DataPtr from (IN) to (OUT)

=> everything else already fixed with RfC 65633

#### Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"

Specification of EFX

~~~~~

~ [SWS\_Efx\_00355] Efx\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

uint8 Efx\_Debounce\_u8\_u8( boolean X, Efx\_DebounceState\_Type \* State, const Efx\_DebounceParam\_Type \* Param, sint32 dT )

~ [SWS\_Efx\_00376] Efx\_MedianSort: The parameter <InType>\* Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS\_Efx\_00309] Efx\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampCheckActivity(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00307] Efx\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampGetSwitchPos(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00193] Efx\_Array\_Average: Include constant for pointer Input-parameter as like below.

<OutType> Efx\_Array\_Average\_<InTypeMn>\_<OutTypeMn>( const <InType>\* Array, uint16 Count)

## Specification of MFL

~~~~~

~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.

Parameters (in): X Initial value for the input state

Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.

Parameters (in): X1_f32 Initial value for input state

Y1_f32 Initial value for output state

Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)

–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.18 Specification Item SWS_Dlt_00654

Trace References:

none

Content:

If the parameter `DltHeaderUseTimestamp` is set to `TRUE` and `DltGeneralGptChannelRef` is configured, the Dlt module shall call the API `Gpt_GetTimeElapsed()` with the configured channel reference (see `DltGeneralGptChannelRef`) to fetch the elapsed time.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of `AppID` and `ContextID`.
- `Dlt_TxFuncton ()` should be renamed as `Dlt_MainFunction` to be consistent with general style of BSW modules.
- Instead of the GPT (SWS_Dlt_00654), the `StbM` should be used to acquire the log time stamps.
- SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
- No prototype is defined for the `EcuID` call-out function (`DltEcuIdCalloutChoice`).
- According to the usual pattern in BSW modules, `E_OK` should be used instead of introducing a special `DLT_OK` for `Std_ReturnType`.
- Configuration parameters `MaxSwcLogMessageLength` and `MaxSwcTraceMessageLength` lack Dlt prefix.
- `Dlt_MessageOptionsType` is not correctly modeled as a struct (or bitfield).
- Chapter 7 contains many references to non-existent parameters.
- Clarification is required regarding MCNT.

–Last change on issue 76283 comment 6–

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from

the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

—Last change on issue 76283 comment 15—

BW-C-Level:

Application	Specification	Bus
1	3	4

1.19 Specification Item SWS_Dlt_00655

Trace References:

none

Content:

If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:
Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set

to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.20 Specification Item SWS_Dlt_00692

Trace References:

none

Content:

The VERB bit shall be set to '1' if the parameter DltUseExtHeaderInNonVerbVerboseMode is set to TRUE. Else, the VERB bit shall be set to '0'.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered

pairs of ApplID and ContextID.

- Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
- Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
- SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
- No prototype is defined for the EcuID call-out function (DltEculdCalloutChoice).
- According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
- Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
- Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
- Chapter 7 contains many references to non-existent parameters.
- Clarification is required regarding MCNT.

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, de-

scription: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.21 Specification Item SWS_Dlt_00729

Trace References:

none

Content:

Module	Imported Type
ComStack_Types	BufReq_ReturnType
	PduIdType
	PduInfoType
	PduLengthType
	RetryInfoType
Dem	Dem_DTCFormatType
	Dem_EventIdType
	Dem_UdsStatusByteType
Gpt	Gpt_ChannelType
	Gpt_ValueType
NvM	NvM_BlockIdType
StbM	StbM_SynchronizedTimeBaseType
	StbM_TimeStampExtendedType
	StbM_TimeStampType
	StbM_UserDataType
Std_Types	Std_ReturnType
	Std_VersionInfoType

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of AppID and ContextID.
- Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
- Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
- SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
- No prototype is defined for the EcuID call-out function (DltEculdCalloutChoice).
- According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
- Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
- Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
- Chapter 7 contains many references to non-existent parameters.
- Clarification is required regarding MCNT.

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter `DltTxPduUsesTp` to `DltTxPdu`, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter `DltRxPduUsesTp` to `DltRxPdu`, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change `SWS_Dlt_00774` by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter `DltGeneralStbMTimeBaseRef` to `DltGeneral`.

Description: "If `TimeStampSupport` is used the Dlt module shall fetch the time from the `StbM` module by calling `StbM_GetCurrentTime` with the here referenced `StbMSynchronizedTimeBase`."

Dependency: "`DltGeneralTimeStampSupport` is set to TRUE and `DltGeneralGptChannelRef` is not configured"

Append "and `DltGeneralStbMTimeBaseRef` is not configured" to the dependency of `DltGeneralGptChannelRef`.

Add `StbM_GetCurrentTime` to the list of optional interfaces (`SWS_Dlt_00763`).

Add `StbM` to chapter 5.

Change `SWS_Dlt_00654` to: "If the parameter `DltHeaderUseTimestamp` is set to TRUE and `DltGeneralGptChannelRef` is configured, the Dlt module shall call the API `Gpt_GetTimeElapsed()` with the configured channel reference (see `DltGeneralGptChannelRef`) to fetch the elapsed time."

Add new SWS item after `SWS_Dlt_00654`: "If the parameter `DltHeaderUseTimestamp` is set to TRUE and `DltGeneralStbMTimeBaseRef` is configured, the Dlt module shall call the API `StbM_GetCurrentTime()` with the configured time base reference (see `DltGeneralStbMTimeBaseRef`) to fetch the current synchronized time and calculate the elapsed time."

—Last change on issue 76283 comment 15—

BW-C-Level:

Application	Specification	Bus
1	3	4

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.22 Specification Item SWS_Dlt_00754

Trace References:

none

Content:

Service name:	Dlt_TriggerTransmitDlt_TriggerTransmit	
Syntax:	Std_ReturnType Dlt_TriggerTransmit(PduIdType TxPduId, PduInfoType* PduInfoPtr)	
Service ID[hex]:	0x41	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduIdDlt_TriggerTransmit.TxPduId	ID of the SDU that is requested to be transmitted.
Parameters (inout):	PduInfoPtrDlt_TriggerTransmit.PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description:	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.23 Specification Item SWS_Dlt_00756

Trace References:

none

Content:

Service name:	Dlt_TpTxConfirmationDlt_TpTxConfirmation
Syntax:	void Dlt_TpTxConfirmation(PduldType id, Std_ReturnType result)
Service ID[hex]:	0x48

Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDlt_TpTxConfirmation.id	Identification of the transmitted I-PDU.
	resultDlt_TpTxConfirmation.result	Result of the transmission of the I-PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	This function is called after the I-PDU has been transmitted on its network, the result indicates whether the transmission was successful or not.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

+ Dlt_StartOfReception

- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.24 Specification Item SWS_Dlt_00763

Trace References:

none

Content:

API function	Description
Dem_DltGetAllExtendedDataRecords	Gets the data of all extended data records of an event.
Dem_DltGetMostRecentFreezeFrameRecordData	Gets the data of an most recent freeze frame record by event. The OBD-II freeze frame is not returned by this function.
Dem_GetDTCOfEvent	Gets the DTC of an event.
Det_ReportError	Service to report development errors.
Gpt_EnableNotification	Enables the interrupt notification for a channel (relevant in normal mode).
Gpt_StartTimer	Starts a timer channel.
NvM_EraseNvBlock	Service to erase a NV block.
NvM_ReadBlock	Service to copy the data of the NV block to its corresponding RAM block.
NvM_WriteBlock	Service to copy the data of the RAM block to its corresponding NV block.
StbM_GetCurrentTime	Returns a time value (Local Time Base derived from Global Time Base) in standard format.
StbM_GetCurrentTimeExtended	Returns a time value (Local Time Base derived from Global Time Base) in extended format.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT
Document: AUTOSAR_SWS_DiagnosticLogAndTrace
Chapter: All
Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUse-
Timestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt
module shall call the API StbM_GetCurrentTime() with the configured time base
reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized
time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

- RfC #76404: [Det] Clarifications on runtime errors

Problem description:

There are several uncertainties/problems in the SWS DET:

1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
3. Configuration of header files for all three error type callouts are missing.
4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exists, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".

–Last change on issue 76404 comment 13–

Agreed solution:

1.

~change SWS_Det_00181/184/187 such that signatures match the APIs

~Figures 3,5, and 7 to be corrected (return missing)

5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."

6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors

7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no call-out exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.

8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."

—Last change on issue 76404 comment 30—

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control mes-

sage via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.25 Specification Item SWS_Dlt_00774

Trace References:

none

Content:

If the parameter DltGeneralRegisterContextNotification is set to TRUE, every time Dlt_UnregisterContext is called, the Dlt module shall send the Dlt Control Message Get LogInfo containing the provided ApplicationId/ContextId with parameter "status" set to 6. 5.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76283: [CONC_627] [MS3a] [WP-A2] Review finding: Several

Problem description:

Name: Martin Schlodder
Role: Reviewer

Finding in SWS DLT

Document: AUTOSAR_SWS_DiagnosticLogAndTrace

Chapter: All

Page or spec item id: See list below.

Description/Comment/Change proposal:

- SWS_Dlt_00774: The PRS does not define a way to report de-registered pairs of ApplID and ContextID.
 - Dlt_TxFunction () should be renamed as Dlt_MainFunction to be consistent with general style of BSW modules.
 - Instead of the GPT (SWS_Dlt_00654), the StbM should be used to acquire the log time stamps.
 - SWS_Dlt_00279 cannot be implemented. How shall a generator know which hooks to bundle within one context ID?
 - No prototype is defined for the EcuID call-out function (DltEcuIDCalloutChoice).
 - According to the usual pattern in BSW modules, E_OK should be used instead of introducing a special DLT_OK for Std_ReturnType.
 - Configuration parameters MaxSwcLogMessageLength and MaxSwcTraceMessageLength lack Dlt prefix.
 - Dlt_MessageOptionsType is not correctly modeled as a struct (or bitfield).
 - Chapter 7 contains many references to non-existent parameters.
 - Clarification is required regarding MCNT.
-

—Last change on issue 76283 comment 6—

Agreed solution:

~ECUC_Dlt_00812:

Remove the dependency (i.e. delete "Can only be true if DltImplementExtended-Header is true")

Add the following parameter to the container DltProtocol:

Name: DltUseVerboseMode

Description: If this flag, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.

Multiplicity: 1

Type: EcucBooleanParamDef

Modify [SWS_Dlt_00692]

from:

The VERB bit shall be set to 1 if the parameter DltUseExtHeaderInNonVerbMode is set to TRUE. Else, the VERB bit shall be set to 0.

to:

The VERB bit shall be set to 1 if the parameter DltUseVerboseMode is set to TRUE. Else, the VERB bit shall be set to 0.

Add boolean post-build/variant parameter DltTxPduUsesTp to DltTxPdu, description: "If set to TRUE, the PDU is transmitted using the TP API. If FALSE, the IF API is used."

Add boolean post-build/variant parameter DltRxPduUsesTp to DltRxPdu, description: "If set to TRUE, the PDU is received using the TP API. If FALSE, the IF API is used."

Change SWS_Dlt_00774 by replacing "parameter status set to 6" by "parameter status set to 5".

Add new pre-compile reference parameter DltGeneralStbMTimeBaseRef to DltGeneral.

Description: "If TimeStampSupport is used the Dlt module shall fetch the time from the StbM module by calling StbM_GetCurrentTime with the here referenced StbMSynchronizedTimeBase."

Dependency: "DltGeneralTimeStampSupport is set to TRUE and DltGeneralGptChannelRef is not configured"

Append "and DltGeneralStbMTimeBaseRef is not configured" to the dependency of DltGeneralGptChannelRef.

Add StbM_GetCurrentTime to the list of optional interfaces (SWS_Dlt_00763).

Add StbM to chapter 5.

Change SWS_Dlt_00654 to: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralGptChannelRef is configured, the Dlt module shall call the API Gpt_GetTimeElapsed() with the configured channel reference (see DltGeneralGptChannelRef) to fetch the elapsed time."

Add new SWS item after SWS_Dlt_00654: "If the parameter DltHeaderUseTimestamp is set to TRUE and DltGeneralStbMTimeBaseRef is configured, the Dlt

module shall call the API StbM_GetCurrentTime() with the configured time base reference (see DltGeneralStbMTimeBaseRef) to fetch the current synchronized time and calculate the elapsed time."

–Last change on issue 76283 comment 15–

BW-C-Level:

Application	Specification	Bus
1	3	4

1.26 Specification Item SWS_Dlt_91006

Trace References:

none

Content:

Service name:	Dlt_StartOfReceptionDlt_StartOfReception	
Syntax:	BufReq_ReturnType Dlt_StartOfReception(PduIdType id, const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePtr)	
Service ID[hex]:	0x46	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDlt_StartOfReception.id	Identification of the I-PDU.
	infoDlt_StartOfReception.info	Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU reception, and the MetaData related to this PDU. If neither first/single frame data nor Meta Data are available, this parameter is set to NULL_PTR.
	TpSduLengthDlt_StartOfReception.TpSduLength	Total length of the N-SDU to be received.
Parameters (inout):	None	
Parameters (out):	bufferSizePtrDlt_StartOfReception.bufferSizePtr	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size (BS) in the transport protocol module.

Return value:	BufReq_ReturnType	BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr. BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged. BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
Description:	This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSduLength equal to 0.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

–Last change on issue 77978 comment 2–

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

–Last change on issue 77978 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.27 Specification Item SWS_Dlt_91007

Trace References:

none

Content:

Service name:	Dlt_TpRxIndicationDlt_TpRxIndication	
Syntax:	void Dlt_TpRxIndication(PduldType id, Std_ReturnType result)	
Service ID[hex]:	0x45	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDlt_TpRxIndication.id	Identification of the received I-PDU.
	resultDlt_TpRxIndication.result	Result of the reception.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

—Last change on issue 77978 comment 2—

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

—Last change on issue 77978 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

1.28 Specification Item SWS_Dlt_91008

Trace References:

none

Content:

Service name:	Dlt_CopyRxDataDlt_CopyRxData	
Syntax:	BufReq_ReturnType Dlt_CopyRxData(PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr)	
Service ID[hex]:	0x44	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idDlt_CopyRxData.id	Identification of the received I-PDU.
	infoDlt_CopyRxData.info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout):	None	
Parameters (out):	bufferSizePtrDlt_CopyRxData.bufferSizePtr	Available receive buffer after data has been copied.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description:	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining buffer is written to the position indicated by bufferSizePtr.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77978: [Dlt] Incomplete Rx data path

Problem description:

In chapter 8.4 the list of callback functions provided for PduR is insufficient. Only the Rx data path for IF is specified, but the Rx data path for TP is not mentioned.

As it is not explicitly mentioned that Dlt usage is limited to specific BUS systems, I assume that also CAN can be used to send control messages to Dlt.

Log and Trace Protocol Specification chapter 5.3 Services / Commands, [PRS_Dlt_00187] is telling that all control messages consist of:

- standard header (min 4 bytes)
- extended header (10 bytes)
- payload (service ID + contained parameters)

When combining these information, [PRS_Dlt_00194] SetLogLevel has a minimum size of at least $4+10+4+13 = 31$ bytes.

Please explain how Dlt is supposed to receive the SetLogLevel control message via CAN that is only supporting 8 bytes of payload, when only Rx data path via IF is supported.

Additionally ECUC_Dlt_00899 is referencing Dlt_StartOfReception, Dlt_CopyRxData and Dlt_TpRxIndication, but the callbacks are not part of chapter 8.4 Call-back notifications.

—Last change on issue 77978 comment 2—

Agreed solution:

Add the following APIs to chapter 8.4 Call-back notifications.

- + Dlt_StartOfReception
- + Dlt_CopyRxData
- + Dlt_TpRxIndication

For consistency reason, let TO model these three APIs - all the parameters shall be the same as used in other BSW modules using TP APIs.

—Last change on issue 77978 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1