

<b>Document Title</b>	SWS_SAEJ1939DiagnosticCommunicationManager Complete Change Documentation 4.3.0 - 4.3.1
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	695

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.1

## Table of Contents

1	SWS_SAEJ1939DiagnosticCommunicationManager	3
1.1	Specification Item ECUC_J1939Dcm_00066 . . . . .	3
1.2	Specification Item SWS_J1939Dcm_00054 . . . . .	4
1.3	Specification Item SWS_J1939Dcm_00090 . . . . .	5
1.4	Specification Item SWS_J1939Dcm_00103 . . . . .	7
1.5	Specification Item SWS_J1939Dcm_00105 . . . . .	8
1.6	Specification Item SWS_J1939Dcm_00132 . . . . .	21
1.7	Specification Item SWS_J1939Dcm_00134 . . . . .	25
1.8	Specification Item SWS_J1939Dcm_00165 . . . . .	26
1.9	Specification Item SWS_J1939Dcm_00197 . . . . .	27
1.10	Specification Item SWS_J1939Dcm_00198 . . . . .	29
1.11	Specification Item SWS_J1939Dcm_00199 . . . . .	30

# 1 SWS\_SAEJ1939DiagnosticCommunicationManager

## 1.1 Specification Item ECUC\_J1939Dcm\_00066

### Trace References:

none

### Content:

Name	J1939DcmSubElementInDataElementInstanceRefJ1939DcmSubElementInDataElementInstance.J1939DcmSubElementInDataElementInstanceRef		
Description	Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationComposite DataType.		
Multiplicity	1		
Type	Instance reference to [ AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE* ]		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Scope / Dependency			

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77305: [J1939Dcm] J1939DcmSubElementInDataElementInstanceRef (ECUC\_J1939Dcm\_00066)

#### Problem description:

Analogue to DcmSubElementInImplDataElementInstanceRef [ECUC\_Dcm\_00992]  
the given type of instance ref to be corrected

from:

Instance reference to [ AUTOSAR-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE\* ]

to:

Instance reference to APPLICATION-COMPOSITE-ELEMENT-DATAPROTOTYPE

context: ROOT-SW-COMPOSITION-PROTOTYPE SWCOMPONENT-  
PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-P  
ROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOT  
YPE\*

**Agreed solution:**

Update in ECUC\_J1939Dcm\_00066 the instance ref to:  
Instance reference to APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE  
context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-  
PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-  
COMPOSITE-ELEMENT-DATA-PROTOTYPE\*  
–Last change on issue 77305 comment 8–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.2 Specification Item SWS\_J1939Dcm\_00054

**Trace References:**

none

**Content:**

On reception of DM13 command message via call of J1939Dcm\_RxIndication with parameter RxPduld set to the configured Pduld J1939DcmDM13RxPduld and the hold signal set to 'not available', the J1939 Diagnostic Communication Manager shall start timeout supervision and call BswM\_J1939DcmBroadcastStatus with the updated broadcast status information; see also [SWS\_J1939Dcm\_00055], [SWS\_J1939Dcm\_00056], [SWS\_J1939Dcm\_00057], and [SWS\_J1939Dcm\_00058].

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #75130: Unclear usage of DM13 and J1939DcmDM13RxPduld

**Problem description:**

[SWS\_J1939Dcm\_00054]

Why does DM13 use J1939Dcm\_RxIndication and not J1939Dcm\_RequestIndication as for all other diagnostic messages? BTW, DM13 is the only DMx that's using J1939Dcm\_RxIndication instead of J1939Dcm\_RequestIndication.

[SWS\_J1939Dcm\_00054] [SWS\_J1939Dcm\_00134]

The configuration parameter "J1939DcmDM13RxPduld", mentioned in the requirements, is not specified in chapter 10. What's the purpose and how shall it be configured?

In general is it unclear when and for which DMx the containers J1939DcmTxPdu and J1939DcmRxPdu shall be used. In my opinion, J1939DcmRxPdu shall not be needed since all requests are received from Rm by using J1939Dcm\_RequestIndication.

**Agreed solution:**

In [SWS\_J1939Dcm\_00054] [SWS\_J1939Dcm\_00134]

J1939DcmDM13RxPduld should replace to ==>J1939DcmRxPduld

–Last change on issue 75130 comment 2–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

### 1.3 Specification Item SWS\_J1939Dcm\_00090

**Trace References:**

none

**Content:**

J1939Dcm shall use following the following development errors:

Type or error	Relevance Related error code	Value [hex]
API service called with wrong PDU or SDU.	Development J1939DCM_E_INVALID_PDU_SDU_ID	0x01
API function called with a NULL Pointer (refer to [SWS_BSW_00212])	Development J1939DCM_E_PARAM_POINTER	0x11
Dem initialisation failed (refer to [SWS_BSW_00050])	Development J1939DCM_E_INIT_FAILED	0x14
API service used in un-initialized state	Development J1939DCM_E_UNINIT	0x20

Type or error	Relevance Related error code	Value [hex]
Dem_Init used in initialized state	Development J1939DCM_E_REINIT	0x21
API service called with or in a wrong state	Development J1939DCM_E_INVALID_STATE	0x06
API service called with wrong node parameter	Development J1939DCM_E_INVALID_NODE	0x08
API service called with wrong channel parameter	Development J1939DCM_E_INVALID_CHANNEL	0x0B
API service called with wrong PGN parameter	Development J1939DCM_E_INVALID_PGN	0x0D
Buffer too small	Development	J1939DCM_E_BUFFER_TOO_SMALL

#### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

##### Problem description:

Inconsistencies in SWS with semantics of Default errors  
–Last change on issue 59085 comment 26–

##### Agreed solution:

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

##### Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

SWS\_LinIf:

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.4 Specification Item SWS\_J1939Dcm\_00103

**Trace References:**

none

**Content:**

Service name:	J1939Dcm_CopyRxDataJ1939Dcm_CopyRxData	
Syntax:	BufReq_ReturnType J1939Dcm_CopyRxData( PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr )	
Service ID[hex]:	0x44	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idJ1939Dcm_CopyRxData.id	Identification of the received I-PDU.
	infoJ1939Dcm_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):	bufferSizePtrJ1939Dcm_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

## 1.5 Specification Item SWS\_J1939Dcm\_00105

#### Trace References:

none

#### Content:



Service name:	J1939Dcm_CopyTxDataJ1939Dcm_CopyTxData	
Syntax:	BufReq_ReturnType J1939Dcm_CopyTxData( PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr )	
Service ID[hex]:	0x43	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	idJ1939Dcm_CopyTxData.id	Identification of the transmitted I-PDU.
	infoJ1939Dcm_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.
	retryJ1939Dcm_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):	availableDataPtrJ1939Dcm_CopyTxData.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. FrIsoTp) 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->TpDataState is TP_DATA_RETRY. 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\_SomelpXf\_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

## 1.6 Specification Item SWS\_J1939Dcm\_00132

### Trace References:

none

## Content:

API function	Description
BswM_J1939DcmBroadcastStatus	This API tells the BswM the desired communication status of the available networks. The status will typically be activated via COM I-PDU group switches.
Dem_J1939DcmClearDTC	Clears <b>active DTCs</b> the status of all event(s) related to the specified DTC(s), as well as <b>previously active DTCs</b> all associated event memory entries for these event(s).
Dem_J1939DcmFirstDTCwithLampStatus	The function sets the filter to the first applicable DTC for the DM31 response for a specific node.
Dem_J1939DcmGetNextDTCwithLampStatus	Gets the next filtered J1939 DTC for DM31 including current LampStatus.
Dem_J1939DcmGetNextFilteredDTC	Gets the next filtered J1939 DTC.
Dem_J1939DcmGetNextFilteredRatio	Gets the next filtered Ratio.
Dem_J1939DcmGetNextFreezeFrame	Gets next freeze frame data. The function stores the data in the provided DestBuffer.
Dem_J1939DcmGetNextSPNInFreezeFrame	Gets next SPN.
Dem_J1939DcmGetNumberOfFilteredDTC	Gets the number of currently filtered DTCs set by the function Dem_J1939DcmSetDTCFilter.
Dem_J1939DcmReadDiagnosticReadiness1	Service to report the value of Diagnostic Readiness 1 (DM05) computed by the Dem.
Dem_J1939DcmReadDiagnosticReadiness2	Service to report the value of Diagnostic Readiness 2 (DM21) computed by the Dem.
Dem_J1939DcmReadDiagnosticReadiness3	Service to report the value of Diagnostic Readiness 3 (DM26) computed by the Dem.
Dem_J1939DcmSetDTCFilter	The function sets the DTC filter for a specific node and returns the composite lamp status of the filtered DTCs.
Dem_J1939DcmSetFreezeFrameFilter	The function sets the FreezeFrame filter for a specific node.
Dem_J1939DcmSetRatioFilter	The function sets the Ratio filter for a specific node and returns the corresponding Ignition Cycle Counter and General Denominator.
Det_ReportError	Service to report development errors.
J1939Rm_SendAck	Requests transmission of an Acknowledgement PG.
PduR_J1939DcmCancelReceive	Requests cancellation of an ongoing reception of a PDU in a lower layer transport protocol module.
PduR_J1939DcmCancelTransmit	Requests cancellation of an ongoing transmission of a PDU in a lower layer communication module.
PduR_J1939DcmTransmit	Requests transmission of a PDU.

## RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- 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 callout 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 #77466: [Dem] J1939DcmClearDTC

**Problem description:**

For J1939Dcm related interfaces, the parameter clientId was introduced in 4.3

1) the function name Dem\_J1939DcmClearDTC does no longer make sense (i mean keeping DCM in, while the clientId is added as parameter).

2) all the J1939 related interface functions are extended by the clientId, but the extension was done in a totally inconsistent way:

In contrast to all the "normal" client-related API's the clientId parameter is added to the end of the functions parameter list (instead of the beginning):

Example:

```
Std_ReturnType Dem_J1939DcmGetNumberOfFilteredDTC( uint16* NumberOfFilteredDTC, uint8 ClientId)
```

Furthermore in at least one place, the clientId is added somewhere in the middle:

Example:

```
Std_ReturnType Dem_J1939DcmSetDTCTFilter(  
Dem_J1939DcmDTCTStatusFilterType DTCTStatusFilter,  
Dem_DTCTKindType DTCTKind,  
Dem_DTCTOriginType DTCTOrigin,  
uint8 ClientId,  
Dem_J1939DcmLampStatusType* LampStatus  
)
```

3) in J1939DCM spec the following api is described: void J1939Dcm\_DemTriggerOnDTCTStatus( uint32 DTCT, uint8 ClientId )



this does not match to DEM which passes along the dtc and statusold/statusnew.  
–Last change on issue 77466 comment 4–

#### Agreed solution:

SWS\_Dem\_00976:

1) New description : clear the status of all event(s) related to the specified DTC(s), as well as all associated event memory entries for these event(s).

–Last change on issue 77466 comment 11–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.7 Specification Item SWS\_J1939Dcm\_00134

#### Trace References:

none

#### Content:

On reception of DM13 command message via call of J1939Dcm\_RxIndication with parameter RxPduld set to the configured **Pduld** J1939Dcm**DM13RxRx**Pduld and the hold signal set to 'all devices' or to 'devices whose broadcast state has been modified', the J1939 Diagnostic Communication Manager shall restart timeout supervision.

#### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75130: Unclear usage of DM13 and J1939DcmDM13RxPduld

#### Problem description:

[SWS\_J1939Dcm\_00054]

Why does DM13 use J1939Dcm\_RxIndication and not J1939Dcm\_RequestIndication as for all other diagnostic messages? BTW, DM13 is the only DMx that's using J1939Dcm\_RxIndication instead of J1939Dcm\_RequestIndication.

[SWS\_J1939Dcm\_00054] [SWS\_J1939Dcm\_00134]

The configuration parameter "J1939DcmDM13RxPduld", mentioned in the requirements, is not specified in chapter 10. What's the purpose and how shall it be configured?

In general is it unclear when and for which DMx the containers J1939DcmTxPdu and J1939DcmRxPdu shall be used. In my opinion, J1939DcmRxPdu shall not be needed since all requests are received from Rm by using J1939Dcm\_RequestIndication.

**Agreed solution:**

In [SWS\_J1939Dcm\_00054] [SWS\_J1939Dcm\_00134]  
J1939DcmDM13RxPduld should replace to ==>J1939DcmRxPduld  
–Last change on issue 75130 comment 2–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

## 1.8 Specification Item SWS\_J1939Dcm\_00165

**Trace References:**

none

**Content:**

If the configuration parameter J1939DcmDevErrorDetect

ECUC\_J1939Dcm\_00003 : is enabled and the return value

return value Dem\_ReturnGetNextFilteredElementType is set to DEM\_FILTERED\_BUFFER\_TOO\_SMALL, the J1939Dcm shall report this error to the Default Error Tracer with the error code J1939DCM\_E\_BUFFER\_TOO\_SMALL, and shall call J1939Rm\_SendAck with parameters 'ackCode' set to J1939RM\_ACK\_NEGATIVE to send a negative acknowledgement (NACK) (considering [SWS\_J1939Dcm\_00113]).

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors  
–Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

**\*\*\* BSW UML Model \*\*\*****SWS\_CanNm:**  
\_\_\_\_\_

## Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRunTimeError

**SWS\_LinIf:**  
\_\_\_\_\_

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

**SWS\_UdpNm:**  
\_\_\_\_\_

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_&lt;Instance Id&gt; (SWS\_UdpNm\_00234)

**\*\*\* ECUC XML \*\*\***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.9 Specification Item SWS\_J1939Dcm\_00197

**Trace References:**[SRS\\_Diag\\_04113](#)**Content:**

When `PduR_J1939DcmTransmit` returns anything but `E_OK`, or when `J1939Dcm_TpTxConfirmation` or `J1939Dcm_TxConfirmation` reports `E_NOT_OK`, the `J1939Dcm` shall retry the failed transmission from the next main function.

#### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77274: [J1939Dcm] Handling of concurrent multi-packet transmission requests from same node

##### Problem description:

According to the SAE specifications, only one multi-packet BAM (i.e. global destination) can be sent from an originator at a given time.

In the J1939Tp SWS, `SWS_J1939Tp_00030` specifies that a transmission request shall be rejected (return `E_NOT_OK`) if the channel is occupied (e.g. a transmission request for a multi-packet BAM while another is on-going from the same SA).

However, the J1939Dcm SWS (an example of an upper layer multi-packet BAM transmission requestor) doesn't specify how the module shall behave if `E_NOT_OK` is returned (e.g. retry transmission request in next main function cycle).

On the other hand, it should be possible to queue transmission requests in J1939Tp (to avoid each upper layer module implementing a retry feature).

A solution for handling such a scenario (and others similar) should be specified

##### Agreed solution:

Add a new section "Failed message transmission" before section 7.3.2 "Termination of message", containing the following new SWS item:

[SWS\_J1939Dcm\_00xxx] /When `PduR_J1939DcmTransmit` returns anything but `E_OK`, or when `J1939Dcm_TpTxConfirmation` or `J1939Dcm_TxConfirmation` reports `E_NOT_OK`, the `J1939Dcm` shall retry the failed transmission from the next main function./(SRS\_Diag\_04113)

–Last change on issue 77274 comment 15–

##### BW-C-Level:

Application	Specification	Bus
1	4	4

## 1.10 Specification Item SWS\_J1939Dcm\_00198

### Trace References:

none

### Content:

Table of runtime errors used by the J1939Dcm module:

Type of error	Related error code	Value [hex]
Buffer too small	J1939DCM_E_BUFFER_TOO_SMALL	0x0E

### RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

#### Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

#### Agreed solution:

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

#### Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRuntimeError

SWS\_LinIf:

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.11 Specification Item SWS\_J1939Dcm\_00199

**Trace References:**

none

**Content:**

API function	Description
Det_ReportRuntimeError	Service to report runtime errors. If a callout has been configured then this callout shall be called.

**RfCs affecting this spec item between releases 4.3.0 and 4.3.1:**

- RfC #59085: Rollout of 'Runtime errors'

**Problem description:**

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

**Agreed solution:**

solution in Column "G" of the new attachment  
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

**Notes:**

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is

referring to that error and adapt them if needed.

- The review task of the ITs shall be done by the WP to which the specification "belongs".

\*\*\* BSW UML Model \*\*\*

SWS\_CanNm:

---

Chapter 8.6.1 Optional Interfaces:

Add within SWS\_CanNm\_00325 the API function Det\_ReportRuntimeError

SWS\_LinIf:

---

SWS\_LinIf\_00359: add Det\_ReportRuntimeError

SWS\_UdpNm:

---

Replace UDPNM\_E\_NO\_INIT with UDPNM\_E\_UNINIT in description of API UdpNm\_MainFunction\_<Instance Id> (SWS\_UdpNm\_00234)

\*\*\* ECUC XML \*\*\*

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

**BW-C-Level:**

Application	Specification	Bus
1	4	1