

<b>Document Title</b>	SWS_SocketAdaptor: 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_SocketAdaptor	4
1.1	Specification Item ECUC_SoAd_00126	4
1.2	Specification Item ECUC_SoAd_00127	5
1.3	Specification Item SWS_SoAd_00073	6
1.4	Specification Item SWS_SoAd_00097	7
1.5	Specification Item SWS_SoAd_00101	20
1.6	Specification Item SWS_SoAd_00137	21
1.7	Specification Item SWS_SoAd_00139	34
1.8	Specification Item SWS_SoAd_00213	36
1.9	Specification Item SWS_SoAd_00224	37
1.10	Specification Item SWS_SoAd_00264	38
1.11	Specification Item SWS_SoAd_00269	40
1.12	Specification Item SWS_SoAd_00272	41
1.13	Specification Item SWS_SoAd_00274	42
1.14	Specification Item SWS_SoAd_00276	44
1.15	Specification Item SWS_SoAd_00279	45
1.16	Specification Item SWS_SoAd_00283	46
1.17	Specification Item SWS_SoAd_00504	48
1.18	Specification Item SWS_SoAd_00523	50
1.19	Specification Item SWS_SoAd_00559	62
1.20	Specification Item SWS_SoAd_00605	64
1.21	Specification Item SWS_SoAd_00607	65
1.22	Specification Item SWS_SoAd_00609	67
1.23	Specification Item SWS_SoAd_00613	68
1.24	Specification Item SWS_SoAd_00615	69
1.25	Specification Item SWS_SoAd_00616	71
1.26	Specification Item SWS_SoAd_00618	72
1.27	Specification Item SWS_SoAd_00621	73
1.28	Specification Item SWS_SoAd_00623	75
1.29	Specification Item SWS_SoAd_00624	76
1.30	Specification Item SWS_SoAd_00627	77
1.31	Specification Item SWS_SoAd_00628	79
1.32	Specification Item SWS_SoAd_00632	80
1.33	Specification Item SWS_SoAd_00655	81
1.34	Specification Item SWS_SoAd_00659	83
1.35	Specification Item SWS_SoAd_00661	85
1.36	Specification Item SWS_SoAd_00672	86
1.37	Specification Item SWS_SoAd_00693	87
1.38	Specification Item SWS_SoAd_00701	89
1.39	Specification Item SWS_SoAd_00705	90

1.40	Specification Item SWS_SoAd_00712	91
1.41	Specification Item SWS_SoAd_00715	93
1.42	Specification Item SWS_SoAd_00718	94
1.43	Specification Item SWS_SoAd_00756	95
1.44	Specification Item SWS_SoAd_00759	97

# 1 SWS\_SocketAdaptor

## 1.1 Specification Item ECUC\_SoAd\_00126

### Trace References:

none

### Content:

Name	SoAdSoConMax		
Description	Specifies the maximum number of SoAd socket connections. Furthermore it defines the platform type used for SoAd_SoConIdType. If SoAdSoConMax is not greater than 256, an uint8 is used, otherwise uint16.		
Multiplicity	1		
Type	EcuIntegerParamDef		
Range	0 .. 65535		
Default value	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: local		

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

- RfC #74087: Change "an uint" to "a uint"

#### Problem description:

Remainder from # 73404:

The affected documents contain text generated artefacts which contain the text "an uint".

Correct is "a uint".

The changes of the artefacts need changes in Metamodel and BSW UML Model.

#### Agreed solution:

Change "an uint" to "a uint" in metamodel artifacts.

–Last change on issue 74087 comment 2–

#### BW-C-Level:

Application	Specification	Bus
1	1	1

## 1.2 Specification Item ECUC\_SoAd\_00127

### Trace References:

none

### Content:

Name	SoAdRoutingGroupMax		
Description	Specifies the maximum number of SoAd routing groups. Furthermore it defines the platform type used for RoutingGroupIdType. If SoAdRoutingGroupMax is not greater than 256, an uint8 is used, otherwise an uint16.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: local		

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

- RfC #74087: Change "an uint" to "a uint"

#### Problem description:

Remainder from # 73404:

The affected documents contain text generated artefacts which contain the text "an uint".

Correct is "a uint".

The changes of the artefacts need changes in Metamodel and BSW UML Model.

#### Agreed solution:

Change "an uint" to "a uint" in metamodel artifacts.

–Last change on issue 74087 comment 2–

**BW-C-Level:**

Application	Specification	Bus
1	1	1

### 1.3 Specification Item SWS\_SoAd\_00073

**Trace References:**

none

**Content:**

The SoAd module shall include the following H-files of other modules:

- Tcplp.h - header file of the AUTOSAR TCP/IP stack
- ComStack\_Types.h [REF]
- Det.h [REF]
- <UL>\_CbK.h (for callback functions of the SoAd upper layer modules)

**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\_SoAd\_00097

**Trace References:**

none

**Content:**

Service name:	SoAd_RxIndicationSoAd_RxIndication
Syntax:	void SoAd_RxIndication( Tcplp_SocketIdType SocketId, const Tcplp_SockAddrType* RemoteAddrPtr, const uint8* BufPtr, uint16 Length )
Service ID[hex]:	0x12
Sync/Async:	Synchronous
Reentrancy:	Reentrant for different SocketIds. Non reentrant for the same SocketId.

Parameters (in):	SocketIdSoAd_RxIndication.SocketId	Socket identifier of the related local socket resource.
	RemoteAddrPtrSoAd_RxIndication.RemoteAddrPtr	Pointer to memory containing IP address and port of the remote host which sent the data.
	BufPtrSoAd_RxIndication.BufPtr	Pointer to the received data.
	LengthSoAd_RxIndication.Length	Data length of the received TCP segment or UDP datagram.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	The TCP/IP stack calls this primitive after the reception of data on a socket. The socket identifier along with configuration information determines which module is to be called.	

**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\_XXXX: 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\_YYYY: 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\_ZZZZ: 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 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 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 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\* Authenticated, uint8\* AuthenticationReqData, uint8\* AuthenticationResData)

To:

Std\_ReturnType <User>\_DoIPRoutingActivationConfirmation(boolean\* Confirmed, const uint8\* ConfirmationReqData, uint8\* ConfirmationResData)

Std\_ReturnType <User>\_DoIPRoutingActivationAuthentication(boolean\* Authenticated, 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\_LdComCbkJCopyTxData\_<sn>  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<sn>( const PduInfoType\* info,  
RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<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

## 1.5 Specification Item SWS\_SoAd\_00101

### Trace References:

none

### Content:

The following table lists development error IDs the SoAd shall use for reporting of development errors to the Default Error Tracer:

Type or error	Relevance Related error code	Value
API service called before initializing the module	Development SOAD_E_NOTINIT UNINIT	0x01
API service called with NULL pointer	Development SOAD_E_PARAM_POINTER	0x02
Invalid argument	Development SOAD_E_INV_ARG	0x03
No buffer space available	Development	SOAD_E_NOBUFS
Unknown PduHeader ID	Development	SOAD_E_INV_PDUHEADER_ID
Invalid PDU ID	Development SOAD_E_INV_PDUID	0x06
Invalid socket address	Development SOAD_E_INV_SOCKETID	0x07
Invalid configuration set selection	Development SOAD_E_INIT_FAILED	0x08
Invalid meta data	Development SOAD_E_INV_METADATA	0x09

### 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

## 1.6 Specification Item SWS\_SoAd\_00137

**Trace References:**

none

**Content:**

Service name:	<Up>_[SoAd][Tp]CopyTxData<Up>_[SoAd][Tp]CopyTxData
---------------	--

Syntax:	BufReq_ReturnType <Up>_[SoAd][Tp]CopyTxData( PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr )	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id<Up>_[SoAd][Tp]CopyTxData.id	Identification of the transmitted I-PDU.
	info<Up>_[SoAd][Tp]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.
	retry<Up>_[SoAd][Tp]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):	availableDataPtr<Up>_[SoAd][Tp]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. 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->TpDataState 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 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\_yyyy, 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\_yyyy, 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\_LdComCbkJCopyTxData\_<sn>  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<sn>( const PduInfoType\* info,  
RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<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

## 1.7 Specification Item SWS\_SoAd\_00139

**Trace References:**

none

**Content:**

Service name:	<Up>_[SoAd][Tp]CopyRxData<Up>_[SoAd][Tp]CopyRxData	
Syntax:	BufReq_ReturnType <Up>_[SoAd][Tp]CopyRxData( PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr )	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	id<Up>_[SoAd][Tp]CopyRxData.id	Identification of the received I-PDU.
	info<Up>_[SoAd][Tp]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):	bufferSizePtr<Up>_[SoAd][Tp]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 <b>data buffer</b> 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

## 1.8 Specification Item SWS\_SoAd\_00213

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_IfTransmit() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_IfTransmit() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.9 Specification Item SWS\_SoAd\_00224

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TpTransmit() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TpTransmit() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.10 Specification Item SWS\_SoAd\_00264

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_RxIndication() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_RxIndication() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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\_SoAd\_00269

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TxConfirmation() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TxConfirmation() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.12 Specification Item SWS\_SoAd\_00272

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TcpAccepted() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TcpAccepted() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

### 1.13 Specification Item SWS\_SoAd\_00274

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TcpConnected() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TcpConnected() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.14 Specification Item SWS\_SoAd\_00276

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TcplpEvent() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TcplpEvent() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.15 Specification Item SWS\_SoAd\_00279

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_LocalIpAddrAssignmentChg() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_LocalIpAddrAssignmentChg() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.16 Specification Item SWS\_SoAd\_00283

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_MainFunction() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_MainFunction() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.17 Specification Item SWS\_SoAd\_00504

**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.
Tcplp_Bind	By this API service the TCP/IP stack is requested to bind a UDP or TCP socket to a local resource.
Tcplp_ChangeParameter	By this API service the TCP/IP stack is requested to change a parameter of a socket. E.g. the Nagle algorithm may be controlled by this API.
Tcplp_Close	By this API service the TCP/IP stack is requested to close the socket and release all related resources.
Tcplp_GetCtrlIdx	Tcplp_GetCtrlIdx returns the index of the controller related to LocalAddrId.
Tcplp_GetIpAddr	Obtains the local IP address actually used by LocalAddrId, the netmask and default router
Tcplp_GetPhysAddr	Obtains the physical source address used by the EthIf controller implicitly specified via LocalAddrId.
Tcplp_GetRemotePhysAddr	Tcplp_GetRemotePhysAddr queries the IP/physical address translation table specified by CtrlIdx and returns the physical address related to the IP address specified by IpAddrPtr. In case no physical address can be retrieved and parameter init Res is TRUE, address resolution for the specified IP address is initiated on the local network.
Tcplp_ReleasIpAddrAssignment	By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be released.
Tcplp_RequestComMode	By this API service the TCP/IP stack is requested to change the Tcplp state of the communication network identified by EthIf controller index.
Tcplp_RequestIpAddrAssignment	By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be initiated.
Tcplp_SoAdGetSocket	By this API service the TCP/IP stack is requested to allocate a new socket. Note: Each accepted incoming TCP connection also allocates a socket resource.
Tcplp_TcpConnect	By this API service the TCP/IP stack is requested to establish a TCP connection to the configured peer.

API function	Description
Tcplp_TcpListen	By this API service the TCP/IP stack is requested to listen on the TCP socket specified by the socket identifier.
Tcplp_TcpReceived	By this API service the reception of socket data is confirmed to the TCP/IP stack.
Tcplp_TcpTransmit	This service requests transmission of data via TCP to a remote node. The transmission of the data is decoupled. Note: The TCP segment(s) are sent dependent on runtime factors (e.g. receive window) and configuration parameter (e.g. Nagle algorithm) .
Tcplp_UdpTransmit	This service transmits data via UDP to a remote node. The transmission of the data is immediately performed with this function call by forwarding it to EthIf.

### 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

## 1.18 Specification Item SWS\_SoAd\_00523

**Trace References:**

none

**Content:**

Service name:	SoAd_CopyTxDataSoAd_CopyTxData	
Syntax:	BufReq_ReturnType SoAd_CopyTxData( Tcplp_SocketIdType SocketId, uint8* BufPtr, uint16 BufLength )	
Service ID[hex]:	0x13	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
Parameters (in):	SocketIdSoAd_CopyTxData.SocketId	Socket identifier of the related local socket resource.
	BufPtrSoAd_CopyTxData.BufPtr	Pointer to buffer for transmission data.
	BufLengthSoAd_CopyTxData.BufLength	Length of provided data buffer.
Parameters (inout):	None	

**None**

Parameters (out):	BufPtrSoAd_CopyTxData.BufPtr	Pointer to buffer for transmission data.
Return value:	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_NOT_OK: Data has not been copied. Request failed. (No further action for Tcplp required. Later the upper layer might either close the socket or retry the transmit request)

Description:	This service requests to copy data for transmission to the buffer indicated. This call is triggered by Tcplp_Transmit(). Note: The call to <Up>_CopyTxData() may happen in the context of Tcplp_Transmit().
--------------	---

**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\_XXXX: 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 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 Parameters (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\* Authenticated, 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\_LdComCbkJCopyTxData\_<sn>  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<sn>( const PduInfoType\* info, RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to  
BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_<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

## 1.19 Specification Item SWS\_SoAd\_00559

**Trace References:**

none

**Content:**

In case PDU header option is enabled (SoAdPduHeaderEnable is TRUE) for a socket connection, SoAd shall convert the message into a PDU within SoAd\_RxIndication() according to the following:

1. assemble the PDU Header into a SoAd receive buffer if it is fragmented in multiple TCP segments
2. extract the PDU Header from the received message
3. select the related socket route according to the received PDU Header ID (So AdRXPduHeaderId); if no socket route can be found, simply discard the PDU and **if development error detection is enabled: raise the runtime error SOAD\_E\_INV\_PDUHEADER\_ID.**
4. use the length field of the PDU Header to identify the length of the actual PDU and the start of the next PDU to proceed with (2) until the end of the message is reached. If the remainder is smaller than a PDU Header or the indicated length within the header SoAd shall stop processing and ignore the rest of the message.

**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

## 1.20 Specification Item SWS\_SoAd\_00605

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TpCancelTransmit() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TpCancelTransmit() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.21 Specification Item SWS\_SoAd\_00607

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_TpCancelReceive() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_TpCancelReceive() shall raise the development error SOAD\_E\_**NOTINITUNINIT**.

**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

## 1.22 Specification Item SWS\_SoAd\_00609

### Trace References:

none

### Content:

If development error detection is enabled: SoAd\_GetSoConId() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_GetSoConId() shall raise the development error SOAD\_E\_NOTINITUNINIT.

### 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

### 1.23 Specification Item SWS\_SoAd\_00613

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_RequestIpAddrAssignment() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_RequestIpAddrAssignment() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.24 Specification Item SWS\_SoAd\_00615

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_OpenSoCon() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_OpenSoCon() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.25 Specification Item SWS\_SoAd\_00616

### Trace References:

none

### Content:

If development error detection is enabled: SoAd\_CloseSoCon() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_CloseSoCon() shall raise the development error SOAD\_E\_NOTINITUNINIT.

### 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

## 1.26 Specification Item SWS\_SoAd\_00618

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_ReleaselpAddrAssignment() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_ReleaselpAddr Assignment() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.27 Specification Item SWS\_SoAd\_00621

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_GetLocalAddr() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_GetLocalAddr() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.28 Specification Item SWS\_SoAd\_00623

### Trace References:

none

### Content:

If development error detection is enabled: SoAd\_GetPhysAddr() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_GetPhysAddr() shall raise the development error SOAD\_E\_NOTINITUNINIT.

### 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

## 1.29 Specification Item SWS\_SoAd\_00624

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_EnableRouting() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_EnableRouting() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

### 1.30 Specification Item SWS\_SoAd\_00627

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_DisableRouting() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_DisableRouting() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.31 Specification Item SWS\_SoAd\_00628

### Trace References:

none

### Content:

If development error detection is enabled: SoAd\_SetRemoteAddr() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_SetRemoteAddr() shall raise the development error SOAD\_E\_NOTINITUNINIT.

### 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

## 1.32 Specification Item SWS\_SoAd\_00632

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_CopyTxData() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_CopyTxData() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

### 1.33 Specification Item SWS\_SoAd\_00655

**Trace References:**

none

**Content:**

Service name:	SoAd_GetRemoteAddrSoAd_GetRemoteAddr
Syntax:	Std_ReturnType SoAd_GetRemoteAddr( SoAd_SoConIdType SoConId, TcpIp_SockAddrType* IpAddrPtr )
Service ID[hex]:	0x1C
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	SoConIdSoAd_GetRemoteAddr.SoConId	socket connection index representing the SoAd socket connection for which the actually specified remote address shall be obtained.
Parameters (inout):	None	
Parameters (out):	IpAddrPtrSoAd_GetRemoteAddr.IpAddrPtr	Pointer to a struct where the retrieved remote address (IP address and port) is stored.
Return value:	Std_ReturnType	Result of operation E_OK The request was successful E_NOT_OK The request was not successful.
Description:	Retrieves the remote address (IP address and port) actually used for the SoAd socket connection specified by SoConId	

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

- RfC #68823: [Sd] Switching of EventHandler from Unicast to Multicast.

#### Problem description:

If the SdEventHandlerMulticastThreshold is set to say 3, and at the current moment two Consumed eventgroups (say A and B) are subscribed to an EventHandler These are being served by Unicast as per [SWS\_SD\_00455], the SubscribeEventGroupAck having previously being sent to these consumed eventgroups.

Now if a new SubscribeEventGroup (C) is received, the Event Handler will shift these three to multicast as per [SWS\_SD\_00455].

Will it also send a new SubscribeEventGroupAck to the Consumed eventgroups A and B to indicate that the events are now sent over multicast?

If so, what are the actions on the client side for the consumed eventgroups A and B?

#### Agreed solution:

~SWS\_SD\_00455

"SoAd\_DisableSpecific Routing" -> "SoAd\_DisableSpecificRouting"

~Figure 9.6

"SoAd\_GetRemoteAddr(return, SoConId, IpAddrPtr)" ->  
 "SoAd\_GetRemoteAddr(return, SdMulticastEventSoConRef, IpAddrPtr)"

~Figure 9.8

"SoAd\_EnableRouting(SdEventHandlerTcp->SdEventActivationRef)" ->  
 "SoAd\_EnableSpecificRouting(SdEventHandlerTcp->SdEventActivationRef, SoConId)"

~Alt [MulticastThreshhold==0 || numOfSubs<MulticastThreshhold]  
 "SoAd\_EnableRouting(SdEventHandlerUdp->SdEventActivationRef)" →  
 "SoAd\_EnableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, So-  
 ConId)"

~Alt [numOfSubs=MulticastThreshhold]  
 "[numOfSubs=MulticastThreshhold]" → "[numOfSubs==MulticastThreshhold]"  
 "SoAd\_EnableRouting(SdEventHandlerMulticast->SdEventActivationRef)" →  
 "SoAd\_EnableSpecificRouting(SdEventHandlerMulticast->SdEventActivationRef,  
 SdEventHandlerMulticast->SdMulticastEventSoConRef)"  
 "SoAd\_DisableRouting(SdEventHandlerUdp->SdEventActivationRef)" →  
 "SoAd\_DisableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef,  
 SoConId)"

-Alt [numOfSubs>=MulticastThreshhold]

~ECUC\_SD\_00096  
 Add at end of description: "This is usually equal to the SdEventActivationRef  
 referenced by SdEventHandlerUdp".

~ECUC\_SD\_00106  
 Add at end of description: "This is usually equal to the SdConsumedEventGroupUd-  
 pActivationRef".

~ECUC\_SD\_00118  
 Change multiplicity of SdMulticastEventSoConRef from optional to mandatory (0..1  
 to 1).  
 –Last change on issue 68823 comment 44–

**BW-C-Level:**

Application	Specification	Bus
1	3	1

### 1.34 Specification Item SWS\_SoAd\_00659

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_GetRemoteAddr() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_GetRemoteAddr() shall raise the development error SOAD\_E\_NOTINITUNINIT.

### 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

### 1.35 Specification Item SWS\_SoAd\_00661

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_IfRoutingGroupTransmit() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_IfRoutingGroupTransmit() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.36 Specification Item SWS\_SoAd\_00672

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_SetUniqueRemoteAddr() shall check that the service SoAd\_Init() was previously called. If the check fails, SoAd\_SetUniqueRemoteAddr() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

### 1.37 Specification Item SWS\_SoAd\_00693

**Trace References:**

none

**Content:**

If **development error detection is enabled**: Whenever a PDU or a part of a PDU is received, that has to be stored in a SoAd receive buffer, is larger than the remaining available buffer size SoAd shall raise the **development runtime** error SOAD\_E\_NOBUFS.

**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

### 1.38 Specification Item SWS\_SoAd\_00701

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_ReadDhcpHostNameOption() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_ReadDhcpHostNameOption() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

### 1.39 Specification Item SWS\_SoAd\_00705

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_WriteDhcpHostNameOption() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_WriteDhcpHostNameOption() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.40 Specification Item SWS\_SoAd\_00712

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_IfSpecificRoutingGroupTransmit() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_IfSpecificRoutingGroupTransmit() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.41 Specification Item SWS\_SoAd\_00715

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_EnableSpecificRouting() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_EnableSpecificRouting() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.42 Specification Item SWS\_SoAd\_00718

**Trace References:**

none

**Content:**

If development error detection is enabled: SoAd\_DisableSpecificRouting() shall check that the service SoAd\_Init was previously called. If the check fails, SoAd\_DisableSpecificRouting() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.43 Specification Item SWS\_SoAd\_00756

**Trace References:**

SRS\_Eth\_00131

**Content:**

If development error detection is enabled: SoAd\_GetAndResetMeasurementData() shall check that the service SoAd\_Init() was previously called. If the check fails, SoAd\_GetAndResetMeasurementData() shall raise the development error SOAD\_E\_NOTINITUNINIT.

**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

## 1.44 Specification Item SWS\_SoAd\_00759

**Trace References:**

none

**Content:**

The following table lists runtime error IDs the SoAd shall use for reporting of runtime errors to the Default Error Tracer:

Type or error	Related error code	Value
No buffer space available	SOAD_E_NOBUFS	0x04
Unknown PduHeader ID	SOAD_E_INV_PDUHEADER_ID	0x05

**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