

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

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

Table of Contents

1	SWS_EFXLibrary	3
1.1	Specification Item SWS_Efx_00193	3
1.2	Specification Item SWS_Efx_00307	15
1.3	Specification Item SWS_Efx_00309	27
1.4	Specification Item SWS_Efx_00355	39
1.5	Specification Item SWS_Efx_00376	52

1 SWS_EFXLibrary

1.1 Specification Item SWS_Efx_00193

Trace References:

none

Content:

Service name:	Efx_Array_Average_<InTypeMn>_<OutTypeMn>Efx_Array_Average	
Syntax:	<pre><OutType> Efx_Array_Average_<InTypeMn>_<OutTypeMn>(const <InType>* Array, uint16 Count)</pre>	
Service ID[hex]:	0x60 and 0x61	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ArrayEfx_Array_Average.Array	Pointer to an array
	CountEfx_Array_Average.Count	Number of array elements
Parameters (inout):	None	
Parameters (out):	None	
Return value:	<OutType>	Return value of the function
Description:	The routine returns average value of an array.	

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\_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 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 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\* 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\_&lt;sn&gt;

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

~~~~~

=&gt; 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   |

## 1.2 Specification Item SWS\_Efx\_00307

**Trace References:**

none

**Content:**

|                     |                                                                            |                                |
|---------------------|----------------------------------------------------------------------------|--------------------------------|
| Service name:       | Efx_RampGetSwitchPosEfx_RampGetSwitchPos                                   |                                |
| Syntax:             | boolean Efx_RampGetSwitchPos(<br>const Efx_StateRamp_Type* State_cpst<br>) |                                |
| Service ID[hex]:    | 0x98                                                                       |                                |
| Sync/Async:         | Synchronous                                                                |                                |
| Reentrancy:         | Reentrant                                                                  |                                |
| Parameters (in):    | State_cpstEfx_RampGetSwitchPos.State_cpst                                  | Pointer to the state structure |
| Parameters (inout): | None                                                                       |                                |
| Parameters (out):   | None                                                                       |                                |
| Return value:       | boolean                                                                    | return value TRUE or FALSE     |
| Description:        | Gets the current switch position of ramp switch function.                  |                                |

**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\_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_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\_SomeIpXf\_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_FrIf_05073 from
FrIf_NumOfStartupFramesPtr (IN)
to
FrIf_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* 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

1.3 Specification Item SWS_Efx_00309

Trace References:

none

Content:

Service name:	Efx_RampCheckActivityEfx_RampCheckActivity	
Syntax:	boolean Efx_RampCheckActivity(const Efx_StateRamp_Type* State_cpst)	
Service ID[hex]:	0x99	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	State_cpstEfx_RampCheck Activity.State_cpst	Pointer to the state structure
Parameters (inout):	None	
Parameters (out):	None	
Return value:	boolean	return value TRUE or FALSE
Description:	This routine checks the status of the ramp and returns TRUE if the ramp is active, otherwise it returns FALSE.	

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\_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 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* 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

1.4 Specification Item SWS_Efx_00355

Trace References:

none

Content:

Service name:	Efx_Debounce_u8_u8Efx_Debounce_u8_u8	
Syntax:	<pre>uint8 boolean Efx_Debounce_u8_u8(boolean X, Efx_DebounceState_Type * State, const Efx_DebounceParam_Type * Param, sint32 dT)</pre>	
Service ID[hex]:	0xB0	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	XEfx_Debounce_u8_u8.X	Input value
	ParamEfx_Debounce_u8_u8.Param	Pointer to state structure of type Efx_DebounceParam_Type
	dTEfx_Debounce_u8_u8.dT	Sample Time
Parameters (inout):	StateEfx_Debounce_u8_u8.State	Pointer to state structure of type Efx_DebounceState_Type
Parameters (out):	None	
Return value:	uint8 boolean	Returns the debounced input value
Description:	This routine debounces a digital input signal and returns the state of the signal as a boolean value.	

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]).

Specification of SOME/IP Transformer

~~~~~

In SWS\_SomeIpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

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

<paramtype> is derived from <type> according to the parameter passing rules 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\_SomeIpXf\_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 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 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

- RfC #77873: [Efx]Return type mismatch in Efx_Debounce_u8_u8

Problem description:

From the requirement [SWS_Efx_00355] the return type of the API Efx_Debounce_u8_u8 is given as uint8 where the function actually returns the variable of boolean type as per the requirement [SWS_Efx_00359]. So this leads to wrong return type of the API.

Agreed solution:

Syntax in [SWS_Efx_00355] shall be changed as below

```
Service name: Efx_Debounce_u8_u8
Syntax: boolean Efx_Debounce_u8_u8(
boolean X,
Efx_DebounceState_Type * State,
Efx_DebounceParam_Type * Param,
sint32 dT
)
```

Service ID[hex]: 0xB0
 Sync/Async: Synchronous
 Reentrancy: Reentrant
 Parameters (in): X Input value
 Param Pointer to state structure of type Efx_DebounceParam_Type
 dT Sample Time
 Parameters (inout): State Pointer to state structure of type Efx_DebounceState_Type
 Parameters (out): None
 Return value: boolean Returns the debounced input value
 Description: This routine debounces a digital input signal and returns the state of the signal as a boolean value.
 –Last change on issue 77873 comment 9–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.5 Specification Item SWS_Efx_00376

Trace References:

none

Content:

Service name:	Efx_MedianSort_<InTypeMn>_<OutTypeMn>Efx_MedianSort	
Syntax:	<OutType> Efx_MedianSort_<InTypeMn>_<OutTypeMn>(<InType>* Array, uint8 N)	
Service ID[hex]:	0xC0 to 0xC4, 0xC8	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ArrayEfx_MedianSort.Array	Pointer to an array
NEfx_MedianSort.N	Size of an array	

None

Parameters (inout):	ArrayEfx_MedianSort.Array	Pointer to an array
Parameters (out):	None	
Return value:	<OutType>	Return value of the function
Description:	Sort an array and return its median value	

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 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
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]).

Specification of SOME/IP Transformer

~~~~~

In SWS\_SomeIpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

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

<paramtype> is derived from <type> according to the parameter passing rules 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\_SomeIpXf\_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 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 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