

<b>Document Title</b>	SWS_MFLLibrary: 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_MFLLibrary	3
1.1	Specification Item SWS_Mfl_00037	3
1.2	Specification Item SWS_Mfl_00223	15
1.3	Specification Item SWS_Mfl_00225	27
1.4	Specification Item SWS_Mfl_00246	39
1.5	Specification Item SWS_Mfl_00260	51
1.6	Specification Item SWS_Mfl_00266	63
1.7	Specification Item SWS_Mfl_00285	76

# 1 SWS\_MFLLibrary

## 1.1 Specification Item SWS\_Mfl\_00037

### Trace References:

none

### Content:

Service name:	Mfl_PT1SetStateMfl_PT1SetState	
Syntax:	<pre>void Mfl_PT1SetState( Mfl_StatePT1_Type* State_cpst, float32 X1_f32, float32 Y1_f32 )</pre>	
Service ID[hex]:	0x1B	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	State_cpstMfl_PT1SetState.State_cpst	Pointer to internal state structure
	X1_f32Mfl_PT1SetState.X1_f32	Initial value for input state
Y1_f32Mfl_PT1SetState.Y1_f32	Initial value for output state	
Parameters (inout):	None	

### None

Parameters (out):	State_cpstMfl_PT1SetState.State_cpst	Pointer to internal state structure
Return value:	None	
Description:	The routine initialises internal state variables of a PT1 element.	

### 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 &lt;Mip&gt;\_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 &lt;Mip&gt;\_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 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  
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 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\_&lt;sn&gt;

BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_&lt;sn&gt;( const PduInfoType\* info, RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to

BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_&lt;sn&gt;( 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\_&lt;User:LoTp&gt;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\_Mfl\_00223

**Trace References:**

none

**Content:**

Service name:	Mfl_RampGetSwitchPosMfl_RampGetSwitchPos	
Syntax:	boolean Mfl_RampGetSwitchPos( const Mfl_StateRamp_Type* State_cpst )	
Service ID[hex]:	0x96	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	State_cpstMfl_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\_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 defined by the SRS BSW General (see SRS\_BSW\_XXXXX, SRS\_BSW\_YYYYY, and SRS\_BSW\_ZZZZZ) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

The following paragraph shall then be removed:

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

In SWS\_Xfrm\_00040 change

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

to

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

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

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

In SWS\_Xfrm\_00044 change

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

to

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

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

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

The following paragraph shall then be removed:

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

Speci?cation of SOME/IP Transformer

~~~~~

In SWS\_SomeIpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

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

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

In SWS\_SomelpXf\_00141 change

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

to

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

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

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

The following paragraph shall then be removed:

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

In SWS\_SomelpXf\_00145 change

```
<type> *data_1, ...  
<type> *data_n
```

to

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

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

<paramtype> is derived from <type> according to the parameter passing rules 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 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\_&lt;sn&gt;( const PduInfoType\* info, RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to

BufReq\_ReturnType Rte\_LdComCbkJCopyTxData\_&lt;sn&gt;( 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\_&lt;User:LoTp&gt;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

~~~~~

==&gt; 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)

=&gt; 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\_Mfl\_00225

**Trace References:**

none

**Content:**

Service name:	Mfl_RampCheckActivityMfl_RampCheckActivity	
Syntax:	boolean Mfl_RampCheckActivity( const Mfl_StateRamp_Type* State_cpst )	
Service ID[hex]:	0x97	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	State_cpstMfl_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 a 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\_XXXXX: Input parameters of scalar and enum types shall be passed as a value.

Type: valid

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

Rationale:

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

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

Dependencies: –

Supporting Material: —

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

Type: valid

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

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

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

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

Dependencies: –

Supporting Material: —

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

Type: valid

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

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

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

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

Dependencies: –

Supporting Material: —

### General Specification of Transformers

~~~~~

In SWS\_Xfrm\_00036 change

const <type>\* dataElement

to

<paramtype> dataElement

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

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

In SWS\_Xfrm\_00038 change

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

to

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

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

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

The following paragraph shall then be removed:

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

In SWS\_Xfrm\_00040 change

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

to

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

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

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

In SWS\_Xfrm\_00044 change

```
<type> *data_1, ...  
<type> *data_n
```

to

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

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

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

The following paragraph shall then be removed:

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

## 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\_SomeIpXf\_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

~~~~~

=> 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\_Mfl\_00246

**Trace References:**

none

**Content:**

Service name:	Mfl_HystDeltaRight_f32_u8Mfl_HystDeltaRight	
Syntax:	boolean Mfl_HystDeltaRight_f32_u8( float32 X, float32 Delta, float32 Rsp, const uint8* State )	
Service ID[hex]:	0xA5	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	XMfl_HystDeltaRight.X	Input value
	DeltaMfl_HystDeltaRight.Delta	Left switching point = rsp - delta
	RspMfl_HystDeltaRight.Rsp	Right switching point
	StateMfl_HystDeltaRight.State	Pointer to state value
Parameters (inout):	None	
Parameters (out):	None	
Return value:	boolean	Returns TRUE or FALSE depending of input value and state value
Description:	Hysteresis with right switching point and delta to left switching point	

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

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

**Problem description:**

SWS\_BSW\_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

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

The various transformer SWS documents shall be adapted accordingly.

–Last change on issue 68035 comment 4–

**Agreed solution:**

BSW UML model

\_\_\_\_\_

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

**General Requirements on Basic Software Modules**

~~~~~

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

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

Type: valid

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

Rationale:

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

```
Std_ReturnType <Mip>_SomeFunction(uint8 SomeParameter);
```

Dependencies: –

Supporting Material: —

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

Type: valid

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

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

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

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

Dependencies: —

Supporting Material: —

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

Type: valid

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

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

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

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

Dependencies: —

Supporting Material: —

General Specification of Transformers

~~~~~

In SWS\_Xfrm\_00036 change

const <type>\* dataElement

to

<paramtype> dataElement

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

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

In SWS\_Xfrm\_00038 change

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

[<type> data\_n]

to

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

[<paramtype> data\_n]

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

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

The following paragraph shall then be removed:

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

[SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

In SWS\_Xfrm\_00040 change

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

to

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

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

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

In SWS\_Xfrm\_00044 change

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

to

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

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

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

The following paragraph shall then be removed:

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

#### 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

## 1.5 Specification Item SWS\_Mfl\_00260

**Trace References:**

none

**Content:**

Service name:	Mfl_Debounce_u8_u8Mfl_Debounce
Syntax:	boolean Mfl_Debounce_u8_u8( boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT )
Service ID[hex]:	0xB0
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	XMfl_Debounce.X	Input value
	ParamMfl_Debounce.Param	Pointer to state structure of type Mfl_DebounceState_Type
	dTMfl_Debounce.dT	Sample Time
Parameters (inout):	StateMfl_Debounce.State	Pointer to structure for debouncing state variables
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.	

**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 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 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  
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 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\_LdComCbkCopyTxData\_<sn>  
BufReq\_ReturnType Rte\_LdComCbkCopyTxData\_<sn>( const PduInfoType\* info,  
RetryInfoType\* retry, PduLengthType\* availableDataPtr ) shall be changed to  
BufReq\_ReturnType Rte\_LdComCbkCopyTxData\_<sn>( const PduInfoType\* info,  
const RetryInfoType\* retry, PduLengthType\* availableDataPtr )

#### Specification of Lin

~~~~~

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

#### Specification of PduR

~~~~~

\* PduR\_<User:LoTp>CopyTxData  
add const to "RetryInfoType\* retry"

#### Specification of J1939Nm

~~~~~

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

#### Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

#### Specification of SPIHandlerDriver

~~~~~

==> nothing to change for SWS SPI

#### Specification of SynchronizedTimeBaseManager

~~~~~

"StbM not affected. All issues listed in the WP-A attachment have been already  
implemented by IT 69124 in context of RfC 65633"

#### Specification of Tcplp

~~~~~

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

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

=> everything else already fixed with RfC 65633

### Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS\_FrTSyn\_00064: parameter versioninfo of type Std\_VersionInfoType\* is marked wrongly as IN. Change to OUT"

### Specification of EFX

~~~~~

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

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

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

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

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

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

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

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

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

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

### Specification of MFL

~~~~~

~ [SWS\_Mfl\_00192] Mfl\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl\_Debounce\_u8\_u8( boolean X, Mfl\_DebounceState\_Type\* State, const Mfl\_DebounceParam\_Type\* Param, float32 dT)

~ [SWS\_Mfl\_00266] Mfl\_DebounceInit: The parameter Mfl\_DebounceState\_Type\* State should be Out instead of In parameter as like below.

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

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

~ [SWS\_Mfl\_00246] Mfl\_HystDeltaRight\_f32\_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl\_HystDeltaRight\_f32\_u8( float32 X, float32 Delta, float32 Rsp, const uint8\* State)

~ [SWS\_Mfl\_00285] Mfl\_MedianSort\_f32\_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS\_Mfl\_00037] Mfl\_PT1SetState: The parameter State\_cpst should be Out instead of In parameter as like below.

Parameters (in): X1\_f32 Initial value for input state

Y1\_f32 Initial value for output state

Parameters (out): State\_cpst Pointer to internal state structure

~ [SWS\_Mfl\_00225] Mfl\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

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

~ [SWS\_Mfl\_00223] Mfl\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

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

–Last change on issue 68035 comment 135–

**BW-C-Level:**

Application	Specification	Bus
1	4	1

## 1.6 Specification Item SWS\_Mfl\_00266

**Trace References:**

none

**Content:**

Service name:	Mfl_DebounceInitMfl_DebounceInit	
Syntax:	void Mfl_DebounceInit( Mfl_DebounceState_Type* State, boolean X )	
Service ID[hex]:	0xB1	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	StateMfl_DebounceInit.State	Pointer to structure for debouncing state variables
XMfl_DebounceInit.X	Initial value for the input state	
Parameters (inout):	None	

**None**

Parameters (out):	StateMfl_DebounceInit.State	Pointer to structure for debouncing state variables
Return value:	None	
Description:	This routine call shall stop the debouncing timer.	

**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  
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 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 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

## 1.7 Specification Item SWS\_Mfl\_00285

### Trace References:

none

### Content:

Service name:	Mfl_MedianSort_f32_f32Mfl_MedianSort_f32_f32	
Syntax:	float32 Mfl_MedianSort_f32_f32( float32* Array, uint8 N )	
Service ID[hex]:	0xBB	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ArrayMflNMfl_MedianSort_f32_f32.Array N	Pointer to Size of an array
Parameters (inout):	NMflArrayMfl_MedianSort_f32_f32.N Array	Size of an array
Parameters (inout):	None Pointer to an array	
Parameters (out):	None	
Return value:	float32	Return value of the function
Description:	This routine sorts values of an array in ascending order. Input array passed by the pointer shall have sorted values after this routine call.	

### 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\* 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\_&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