

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

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

Table of Contents

1	SWS_COMBasedTransformer	3
1.1	Specification Item SWS_ComXf_00007	3
1.2	Specification Item SWS_ComXf_00023	15
1.3	Specification Item SWS_ComXf_00036	18

1 SWS_COMBasedTransformer

1.1 Specification Item SWS_ComXf_00007

Trace References:

SRS_Xfrm_00201

Content:

Service name:	ComXf_<transformerId>ComXf_<transformerId>	
Syntax:	uint8 ComXf_<transformerId>(uint8* buffer, uint32* bufferLength, const <typeparamtype> * dataElement)	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	dataElementComXf_<transformerId>.dataElement	Data element which shall be transformed
Parameters (inout):	None	
Parameters (out):	bufferComXf_<transformerId>.buffer	Buffer allocated by the RTE, where the transformed data has to be stored by the transformer
	bufferLengthComXf_<transformerId>.bufferLength	Used length of the buffer
Return value:	uint8	0x00 (E_OK): Serialization successful 0x81 (E_SER_GENERIC_ERROR): A generic error occurred
Description:	This function transforms a Sender/Receiver communication using the serialization of COM Based Transformer. It takes the data element as input and outputs an uint8 array containing the serialized data.	

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 defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

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

In SWS_Xfrm_00040 change

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

to

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

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

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

In SWS_Xfrm_00044 change

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

to

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

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

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

The following paragraph shall then be removed:

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

Speci?cation of SOME/IP Transformer

~~~~~

In SWS\_SomelpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

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

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

In SWS\_SomeIpXf\_00141 change

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

to

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

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

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

The following paragraph shall then be removed:

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



In SWS\_SomelpXf\_00145 change

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

to

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

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

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

The following paragraph shall then be removed:

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

## Specification of COM Based Transformer

~~~~~

In SWS_ComXf_00007 change

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

to

```
<paramtype> dataElement
```

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

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

Specification of Time Sync over Ethernet

~~~~~

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

#### Specification of SWS FlexRay Interface

~~~~~

Change SWS_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_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

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

Problem description:

Remainder from # 73404:

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

Correct is "a uint".

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

Agreed solution:

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

–Last change on issue 74087 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item SWS_ComXf_00023

Trace References:

SRS_Xfrm_00201

Content:

The COM Based Transformer **only supports signal groups** shall support signal group where all group signals are mapped **consecutively successively** (possibly with gaps where positions in the signal group layout have no corresponding signal defined) to the IPdu.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74115: [ComXf]: Limitation SWS_ComXf_00023 can be updated as per Com requirement SWS_Com_00841

Problem description:

As per Bugzilla # 68315

It is concluded that during reception of Signal Group Array the group signals in it may not be consecutive and other signals may be interleaved or gaps can be there.(as per SWS_Com_00841)

But in Com based transformer Specification

There is a limitation

SWS_ComXf_00023: The COM Based Transformer only supports signal groups where all group signals are mapped consecutively to the IPdu.

As per Bugzilla # 68315 during reception of signalgroup, Group signals need not be mapped consecutively.

Hence can we update the mentioned limitation as below

SWS_ComXf_00023: The COM Based Transformer only supports signal groups where all group signals are mapped consecutively to the IPdu on transmission side.

If above mentioned solution is acceptable

During reception: Com based transformer shall get buffer from RTE (RTE shall get data from Com_ReceiveSignalGroupArray), in which along with group signals there may be some other signals data (which are not part of this signal group) or some other data (if gaps are there).

But Com Based transformer shall extract the Groups signals in that signal group based on bit positions configured. Data that is not related to this signal group shall be untouched.

Is this understanding correct? please clarify

Regards,
KPIT

Update:

ComXf should be updated to respect the "ComTxIPduUnusedAreaDefault" value in order to have well defined values for unused bits in the byte array prepared by ComXf. This can be done by referring SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.unusedBitPattern of the respective System Signal Group or ComIPdu:ComTxIPdu:ComTxIPduUnusedAreasDefault parameter.

This approach would allow gaps in the signal group definition (Normal signal and other Signal group shall not be mapped in the gap) and also define the value for the unused bits of signal which are shorter than 8 bits.

–Last change on issue 74115 comment 14–

Agreed solution:

SWS ComXf

- Update SWS_ComXf_00023 to:

SWS_ComXf_00023: The COM Based Transformer shall support signal group where all group signals are mapped successively (possibly with gaps where positions in the signal group layout have no corresponding signal defined) to the IPdu.

- Add before [SWS_ComXf_00014]:

[SWS_ComXf_000x1] If the signal layout of the signal group array representation contains gaps, those gaps shall be set during transmission to the value defined by the ComTxIPduUnusedAreasDefault of the respective ComIPdu that this signal group is mapped to.

Gaps in the signal group array representation may occur because the layout is not fully packed and there are bits (or even bytes) that have no signal defined for.

- Add attached figure for further explanation

COM parameter ComTxIPduUnusedAreasDefault shall be added in "B Used ECU Configuration" along with other COM parameters used by ComXf.

=====

COM SWS

SWS_Com_00841: The UINT8-array based access to signal groups shall only be used if the following preconditions apply:

Uses only fix sized data types for the composite data.

Signal groups, which are mapped byte aligned to the I-PDU.

Signal groups where all group signals are mapped consecutively to the I-PDU on

transmission side.

to

SWS_Com_00841: The UINT8-array based access to signal groups shall only be used if the following preconditions apply:

Uses only fix sized data types for the composite data.

Signal groups, which are mapped byte aligned to the I-PDU.

Signal groups, which are not intermitted by other signals (but may contain gaps).

Remove following Note after SWS_COM_00845

Note: Please note that for reception the signal group may not be consecutive and other signals may be interleaved in the uint8-array representation of the received signal group.

–Last change on issue 74115 comment 27–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SWS_ComXf_00036

Trace References:

SRS_Xfrm_00201, SRS_Xfrm_00202

Content:

If the signal layout of the signal group array representation contains gaps, those gaps shall be set during transmission to the value defined by the ComTxIPdu.ComTxIPduUnusedAreasDefault of the respective ComTxIPdu that this signal group is mapped to.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74115: [ComXf]: Limitation SWS_ComXf_00023 can be updated as per Com requirement SWS_Com_00841

Problem description:

As per Bugzilla # 68315

It is concluded that during reception of Signal Group Array the group signals in it may not be consecutive and other signals may be interleaved or gaps can be

there.(as per SWS_Com_00841)

But in Com based transformer Specification

There is a limitation

SWS_ComXf_00023: The COM Based Transformer only supports signal groups where all group signals are mapped consecutively to the IPdu.

As per Bugzilla # 68315 during reception of signalgroup, Group signals need not be mapped consecutively.

Hence can we update the mentioned limitation as below

SWS_ComXf_00023: The COM Based Transformer only supports signal groups where all group signals are mapped consecutively to the IPdu on transmission side.

If above mentioned solution is acceptable

During reception: Com based transformer shall get buffer from RTE (RTE shall get data from Com_ReceiveSignalGroupArray), in which along with group signals there may be some other signals data (which are not part of this signal group) or some other data (if gaps are there).

But Com Based transformer shall extract the Groups signals in that signal group based on bit positions configured. Data that is not related to this signal group shall be untouched.

Is this understanding correct? please clarify

Regards,
KPIT

Update:

ComXf should be updated to respect the "ComTxIPduUnusedAreaDefault" value in order to have well defined values for unused bits in the byte array prepared by ComXf. This can be done by referring SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.unusedBitPattern of the respective System Signal Group or ComIPdu:ComTxIPdu:ComTxIPduUnusedAreasDefault parameter.

This approach would allow gaps in the signal group definition (Normal signal and other Signal group shall not be mapped in the gap) and also define the value

for the unused bits of signal which are shorter than 8 bits.

–Last change on issue 74115 comment 14–

Agreed solution:

SWS ComXf

- Update SWS_ComXf_00023 to:

SWS_ComXf_00023: The COM Based Transformer shall support signal group where all group signals are mapped successively (possibly with gaps where positions in the signal group layout have no corresponding signal defined) to the IPdu.

- Add before [SWS_ComXf_00014]:

[SWS_ComXf_000x1] If the signal layout of the signal group array representation contains gaps, those gaps shall be set during transmission to the value defined by the ComTxIPduUnusedAreasDefault of the respective ComIPdu that this signal group is mapped to.

Gaps in the signal group array representation may occur because the layout is not fully packed and there are bits (or even bytes) that have no signal defined for.

- Add attached figure for further explanation

COM parameter ComTxIPduUnusedAreasDefault shall be added in "B Used ECU Configuration" along with other COM parameters used by ComXf.

=====

COM SWS

SWS_Com_00841: The UINT8-array based access to signal groups shall only be used if the following preconditions apply:

Uses only fix sized data types for the composite data.

Signal groups, which are mapped byte aligned to the I-PDU.

Signal groups where all group signals are mapped consecutively to the I-PDU on transmission side.

to

SWS_Com_00841: The UINT8-array based access to signal groups shall only be used if the following preconditions apply:

Uses only fix sized data types for the composite data.

Signal groups, which are mapped byte aligned to the I-PDU.

Signal groups, which are not intermitted by other signals (but may contain gaps).

Remove following Note after SWS_COM_00845

Note: Please note that for reception the signal group may not be consecutive and other signals may be interleaved in the uint8-array representation of the received signal group.

–Last change on issue 74115 comment 27–

BW-C-Level:

Application	Specification	Bus
1	1	1