

<b>Document Title</b>	Specification of COM Based Transformer
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	662

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

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Updated handling of gaps in the array representation of a signal group.</li> <li>Clarification on parameter passing.</li> </ul>
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Updated include file structure figure <a href="#">5.1</a>.</li> <li>Clarification on postBuild configuration in chapter <a href="#">10</a>.</li> <li>Added support for unqueued communication when no data is available in [<a href="#">SWS_ComXf_00035</a>].</li> </ul>
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Exclude support for external trigger communication [<a href="#">SWS_ComXf_00032</a>]</li> </ul>
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Initial Release</li> </ul>

## Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Table of Contents

1	Introduction and functional overview	5
2	Acronyms and Abbreviations	6
3	Related documentation	7
3.1	Input documents	7
3.2	Related standards and norms	8
3.3	Related specification	8
4	Constraints and assumptions	9
4.1	Limitations	9
4.2	Applicability to car domains	9
5	Dependencies to other modules	10
5.1	File structure	10
5.1.1	Code file structure	10
5.1.2	Header file structure	10
6	Requirements Tracing	12
7	Functional specification	13
7.1	Specification of the COM Based Transformer	16
7.2	Error classification	19
7.2.1	Development Errors	19
7.2.2	Runtime Errors	20
7.2.3	Transient Faults	20
7.2.4	Production Errors	20
7.2.5	Extended Production Errors	20
8	API specification	21
8.1	Imported types	21
8.2	Type definitions	21
8.3	Function definitions	21
8.3.1	ComXf_<transformerId>	21
8.3.2	ComXf_Inv_<transformerId>	23
8.3.3	ComXf_Init	24
8.3.4	ComXf_DelInit	24
8.3.5	ComXf_GetVersionInfo	25
8.4	Callback notifications	25
8.5	Scheduled functions	25
8.6	Expected interfaces	25
9	Sequence diagrams	26
10	Configuration specification	27

A	Referenced Meta Classes	28
B	Used ECU Configurion	40
C	History of Constraints and Specification Items	64
C.1	Constraint History of this Document according to AUTOSAR R4.2.1 . . .	64
C.2	Constraint History of this Document according to AUTOSAR R4.2.2 . . .	64
C.2.1	Added Traceables in 4.2.2 . . . . .	64
C.2.2	Changed Traceables in 4.2.2 . . . . .	64
C.2.3	Deleted Traceables in 4.2.2 . . . . .	64
C.3	Constraint History of this Document according to AUTOSAR R4.3.0 . . .	64
C.3.1	Added Traceables in 4.3.0 . . . . .	64
C.3.2	Changed Traceables in 4.3.0 . . . . .	64
C.3.3	Deleted Traceables in 4.3.0 . . . . .	65
C.4	Constraint History of this Document according to AUTOSAR R4.3.1 . . .	65
C.4.1	Added Traceables in 4.3.1 . . . . .	65
C.4.2	Changed Traceables in 4.3.1 . . . . .	65
C.4.3	Deleted Traceables in 4.3.1 . . . . .	65

## 1 Introduction and functional overview

The transformer feature provides functionality to chain arbitrary transformers when sending and receiving data in the RTE. The COM Based Transformer provides this functionality when the target bus system uses a fixed communication matrix with packed data representations.

## 2 Acronyms and Abbreviations

No specific terms have been introduced additionally to those already defined in [1].

## 3 Related documentation

### 3.1 Input documents

#### Bibliography

- [1] Glossary  
AUTOSAR\_TR\_Glossary
- [2] General Specification on Transformers  
AUTOSAR\_ASWS\_TransformerGeneral
- [3] Specification of RTE Software  
AUTOSAR\_SWS\_RTE
- [4] Specification of Communication  
AUTOSAR\_SWS\_COM
- [5] General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_BSWGeneral
- [6] Requirements on Transformer  
AUTOSAR\_SRS\_Transformer
- [7] System Template  
AUTOSAR\_TPS\_SystemTemplate
- [8] General Specification of Basic Software Modules  
AUTOSAR\_SWS\_BSWGeneral

## 3.2 Related standards and norms

Not applicable.

## 3.3 Related specification

AUTOSAR provides a General Specification on Transformers [2], which is also valid for COM Based Transformer.

Thus, the specification ASWS Transformer General shall be considered as additional and required specification for COM Based Transformer.



## 4 Constraints and assumptions

### 4.1 Limitations

For the COM Based Transformer all general transformer limitations (see [2]) apply.

Additionally the following restrictions apply for the COM Based Transformer:

**[SWS\_ComXf\_00017]** [ The COM Based Transformer does not support Client-Server communication. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00032]** [ The COM Based Transformer does not support external trigger communication. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00018]** [ The COM Based Transformer only supports composite data types (i.e. Signal Groups from COM). ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00019]** [ The COM Based Transformer only supports fix sized data types. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00022]** [ The COM Based Transformer only supports signal groups which are byte aligned. ]([SRS\\_Xfrm\\_00201](#))

**[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. ]([SRS\\_Xfrm\\_00201](#))

### 4.2 Applicability to car domains

The COM Based Transformer can be used for all domain applications when a fixed communication matrix is used.

## 5 Dependencies to other modules

The AUTOSAR RTE [3] has to exist to execute the COM Based Transformer.

The AUTOSAR COM configuration [4] of the data handled by the COM Based Transformer has to exist in order to allow the configuration of the COM Based Transformer.

### 5.1 File structure

#### 5.1.1 Code file structure

The source code file structure is defined in the [2].

#### 5.1.2 Header file structure

The header file structure of the COM Based Transformer is shown in Figure 5.1.

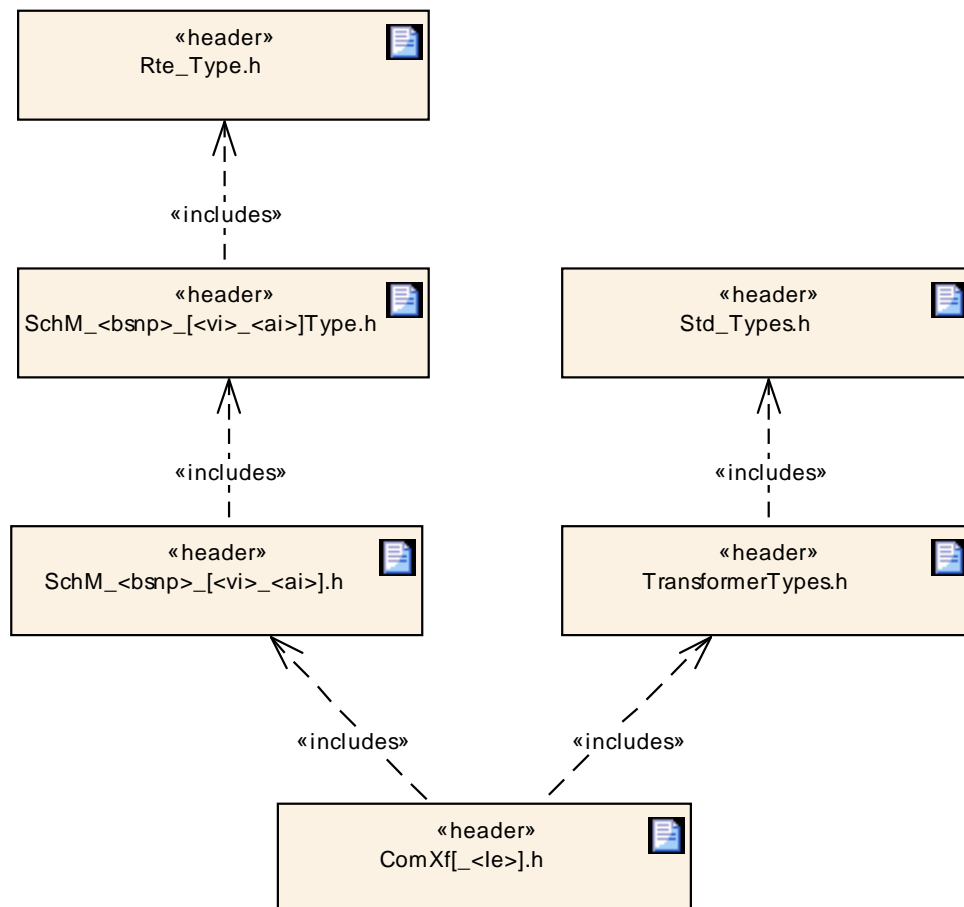


Figure 5.1: Header File Structure of COM Based Transformer

**[SWS\_ComXf\_00001]** [ The header file `ComXf[_<Ie>].h` shall be the main include file for the COM Based transformer and include `TransformerTypes.h` and its Module Interlink Header file `SchM_<bsnp>[_<vi>_<ai>].h` where  
<Ie> is the optional implementation specific file name extension according [SWS\_BSW\_00103],  
<bsnp> is the BSW Scheduler Name Prefix according [SWS\_Rte\_07593] and [SWS\_Rte\_07594],  
<vi> is the [vendorId](#) of the BSW module and  
<ai> is the [vendorApiInfix](#) of the BSW module. ]([SRS\\_BSW\\_00346](#))

The file `TransformerTypes.h` contains the general transformer data types.

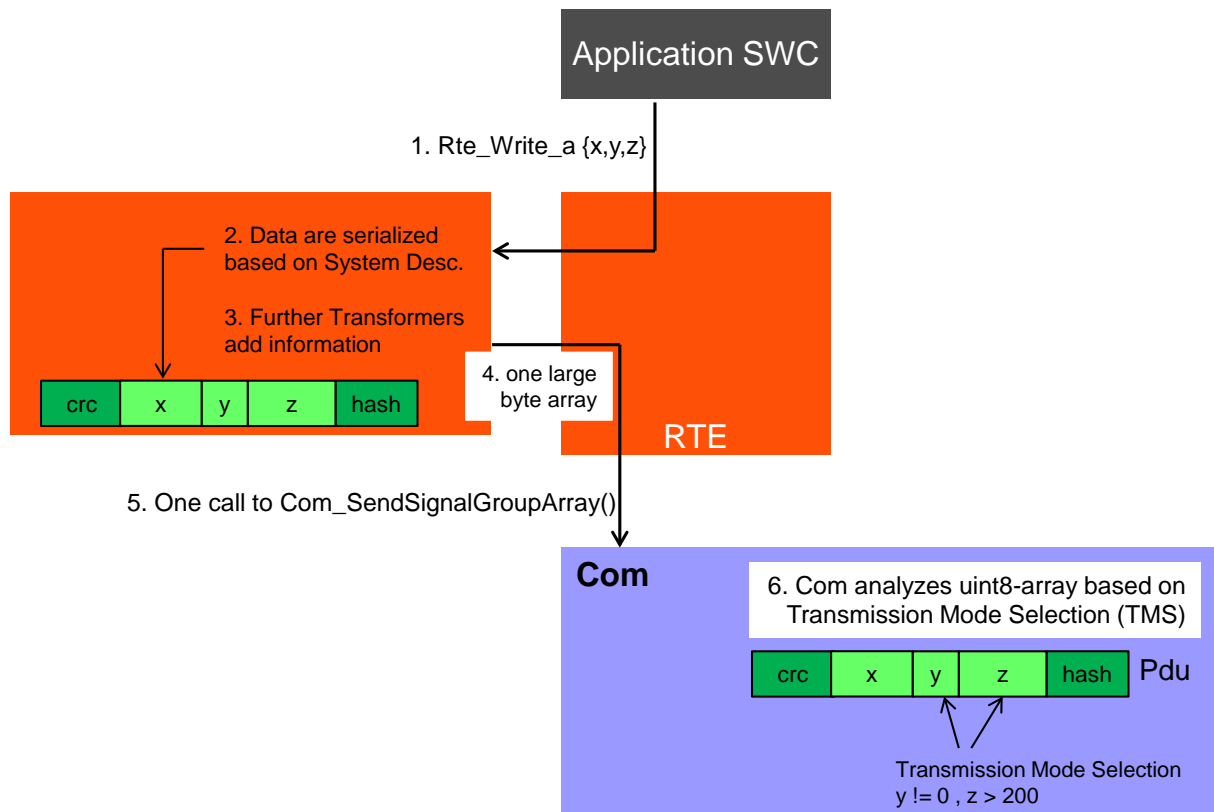
## 6 Requirements Tracing

The following table references the features specified in [5] and [6] and links to the fulfillments of these.

Feature	Description	Satisfied by
[SRS_BSW_00159]	All modules of the AUTOSAR Basic Software shall support a tool based configuration	[SWS_ComXf_00025]
[SRS_BSW_00337]	Classification of development errors	[SWS_ComXf_00028]
[SRS_BSW_00346]	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	[SWS_ComXf_00001]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_ComXf_00030]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_ComXf_00024] [SWS_ComXf_00026] [SWS_ComXf_00027]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_ComXf_00024] [SWS_ComXf_00026] [SWS_ComXf_00027]
[SRS_BSW_00441]	Naming convention for type, macro and function	[SWS_ComXf_00030]
[SRS_Xfrm_00009]	A fixed set of transformer classes shall exist	[SWS_ComXf_00003]
[SRS_Xfrm_00011]	A transformer shall belong to a specific transformer class	[SWS_ComXf_00003] [SWS_ComXf_00006]
[SRS_Xfrm_00201]	The COM Based Transformer shall define the serialization of atomic and structured data elements into linear arrays based on a fixed data mapping	[SWS_ComXf_00004] [SWS_ComXf_00005] [SWS_ComXf_00007] [SWS_ComXf_00008] [SWS_ComXf_00009] [SWS_ComXf_00010] [SWS_ComXf_00011] [SWS_ComXf_00012] [SWS_ComXf_00013] [SWS_ComXf_00014] [SWS_ComXf_00015] [SWS_ComXf_00016] [SWS_ComXf_00017] [SWS_ComXf_00018] [SWS_ComXf_00019] [SWS_ComXf_00020] [SWS_ComXf_00021] [SWS_ComXf_00022] [SWS_ComXf_00023] [SWS_ComXf_00032] [SWS_ComXf_00035] [SWS_ComXf_00036]
[SRS_Xfrm_00202]	The COM Based Transformer shall take its configuration from the COM module	[SWS_ComXf_00005] [SWS_ComXf_00014] [SWS_ComXf_00020] [SWS_ComXf_00025] [SWS_ComXf_00031] [SWS_ComXf_00033] [SWS_ComXf_00034] [SWS_ComXf_00036]

## 7 Functional specification

When a SWC initiates an inter-ECU communication which is configured to be transformed, the SWC hands the data over to the RTE. The RTE executes the configured transformer chain which contains - if the configuration demands this - the COM Based Transformer.

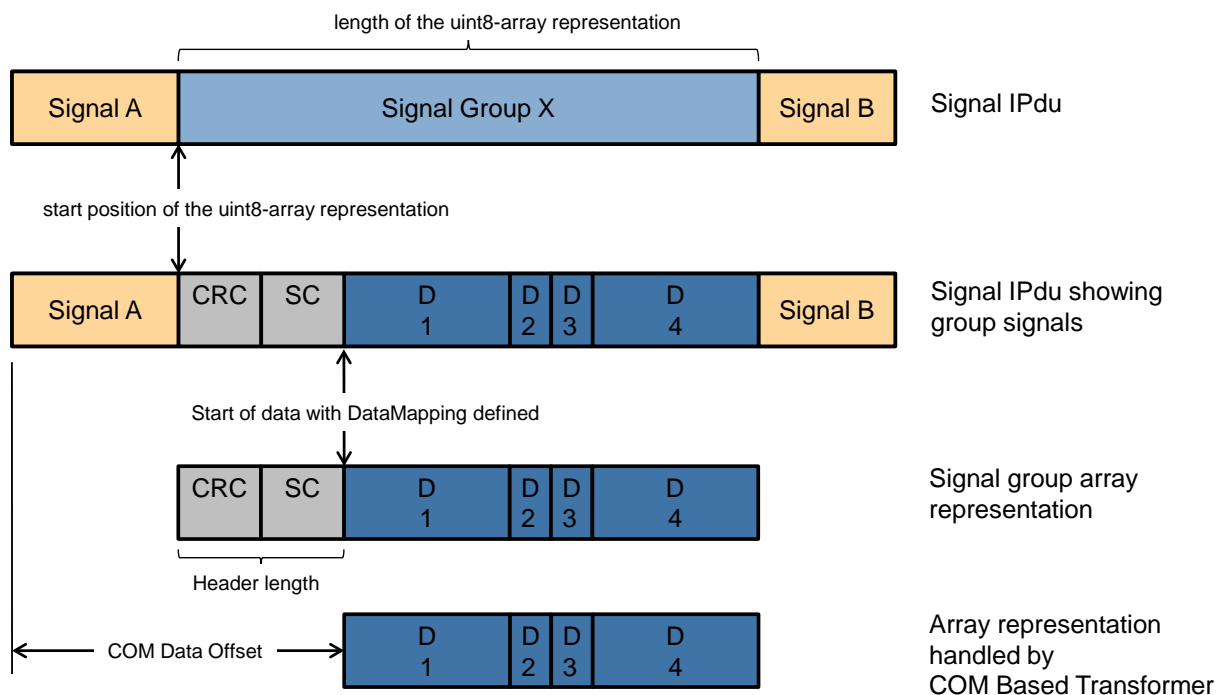


**Figure 7.1: Overview of COM Based Transformer**

The COM Based Transformer on the sender side serializes the data of the SWC and brings them into a uint8-array representation based on the communication matrix description. The uint8-array representation is forwarded to the COM module to be placed inside the respective IPdu. The COM module may analyze (depending on the configuration of the Transmission Mode Selection – TMS) the provided uint8-array and trigger the respective transmission mode. The IPdu is sent via the communication stack over the bus to the receiver(s).

The RTE of the receiver side executes the transformer chain in the reverse order. The COM Based transformer of the receiver deserializes the linear data back into the original data structure. These are handed over to the receiving SWC.

From the SWC’s point of view it is totally transparent whether data are transformed or not.

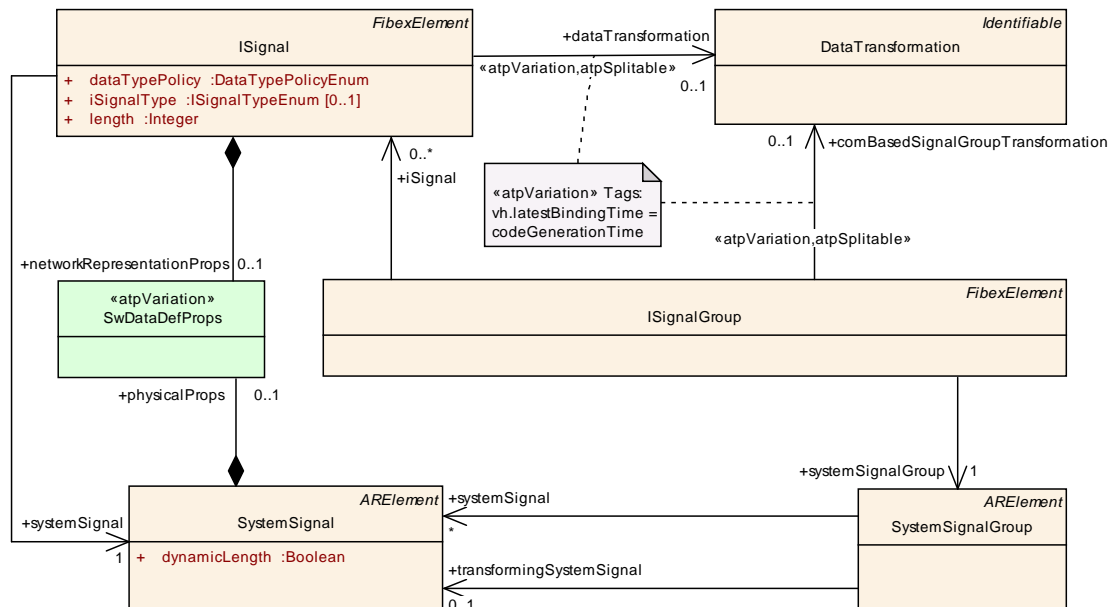


**Figure 7.2: IPdu and signal layout**

The handling of the data inside COM's IPdu buffer and the transformer buffer is shown in figure 7.2.

The `ISignalIPdu` is handled by the COM module as `ComIPdu` and may contain several parts (signals and signal groups). A signal group in COM is represented by the `ComSignalGroup` container.

In the System Template [7] it is possible to define that a signal group shall be handled by the COM Based Transformer. The usage of the COM Based Transformer for a specific transformer chain is defined by the reference `comBasedSignalGroupTransformation` from the `ISignalGroup` to `DataTransformation` (see figure 7.3).



**Figure 7.3: System Template Transformed communication**

In the Ecu configuration of the COM module such `ComSignalGroups` have a `ComSignalGroupArrayAccess=true` parameter defined.

If the `ComSignalGroupArrayAccess=true` then it implicitly defines the length of the signal group ([SWS\_Com\_00845] [4]) and the start position inside the `ComIPdu` where the signal group starts ([SWS\_Com\_00844] [4]). Thus there can be several signals and signal groups defined inside an `ISignalIPdu` (e.g. 'Signal A' and 'Signal B' are part of the IPdu but are not considered by the COM Based Transformer for the 'Signal Group X').

The Com APIs `Com_SendSignalGroupArray` and `Com_ReceiveSignalGroupArray` handle the signal group as array representation based on length of the signal group and the start position inside the `ComIPdu`. This array representation contains all signals that belong to the signal group, regardless whether the application software has a data mapping defined or not.

As an example in figure 7.2 the 'Signal Group X' consists of the signals 'CRC', 'SC', 'D1', 'D2', 'D3', and 'D4'. Thus the RTE will interact with COM based on the whole array representation of this signal group with length of the signal group.

When the RTE interacts with the COM Based Transformer also all the other potential transformers need to be considered in order to determine which part of the array representation of the signal group actually is provided to each transformer since each transformer may add data during sending (or remove data during reception).

E.g. the part of the array representation which holds the 'CRC' and 'SC' signals are handled by the E2E transformer but will not be considered by the COM Based Transformer. The RTE will provide the data buffer from the 'Start of data with DataMapping defined' location to the COM Based Transformer.

## 7.1 Specification of the COM Based Transformer

Serialization describes the way data is represented in protocol data units (PDUs) transported over a network. For the COM Based Transformer the serialization is defined by the communication matrix using the System Template [7]. The communication matrix information is taken over to the Ecu configuration of the COM module.

**[SWS\_ComXf\_00005]** [ The serialization is based on the Ecu configuration of the COM module and

- the Software Components `PortPrototype`,
- the `dataElement` list defined by the respective `SenderReceiverInterface`,
- the `SenderReceiverToSignalGroupMapping`,
- and the `ISignalToIPduMapping` for the `SystemSignalGroup`.

]([SRS\\_Xfrm\\_00201](#), [SRS\\_Xfrm\\_00202](#))

**[SWS\_ComXf\_00003]** [ The COM Based transformer shall only be used as the top-most transformer (first) in a transformer chain. ]([SRS\\_Xfrm\\_00009](#), [SRS\\_Xfrm\\_00011](#))

The COM Based transformer serializes structured data into a linear form. Therefore it can only be used as the first transformer on the sending side and the last transformer on the receiving side.

**[SWS\_ComXf\_00004]** [ The COM Based transformer defined in this document shall be used as a transformer if

- the attribute `protocol` of the `TransformationTechnology` is set to `COMBased`
- and the attribute `version` of the `TransformationTechnology` is set to `1.0.0`
- and the attribute `transformerClass` of the `TransformationTechnology` is set to `serializer`

]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00015]** [ The COM Based Transformer shall support all basic data types that are supported by the COM module in [SWS\_Com\_00675] except for `UINT8_DYN`. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00016]** [ The COM Based Transformer shall handle each `dataElement` of the `SenderReceiverInterface` individually. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00021]** [ The COM Based Transformer shall handle each `dataElement` of the `SenderReceiverInterface` like defined for the COM module [4] when the COM API `Com_SendSignal` (rep. `Com_ReceiveSignal`) is called for a shadow signal. ]([SRS\\_Xfrm\\_00201](#))

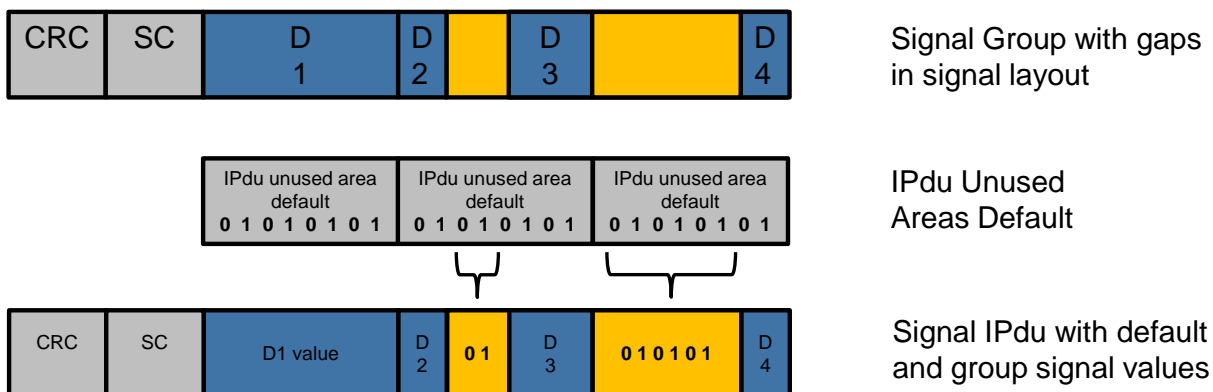


This defines that the COM Based Transformer performs all actions equally to the COM module. This does include functionality like endianness conversion and sign extension.

The COM Configuration implicitly defines the length of the signal group ([SWS\_Com\_00845] [4]) and the start position inside the `ComIPdu` where the signal group starts ([SWS\_Com\_00844] [4]). In order to place the transformed data element into the data buffer provided by the RTE the COM Based Transformer needs to respect the offset introduced by the position of the `ComGroupSignal` inside the `ComIPdu` (defined by the start position inside the `ComIPdu`) and the additional offset introduced by header data which is handled by other transformers called after the COM Based Transformer.

**[SWS\_ComXf\_00036]** [ 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 `ComTxIPdu` that this signal group is mapped to. ]([SRS\\_Xfrm\\_00201](#), [SRS\\_Xfrm\\_00202](#))

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 (see 7.4).



**Figure 7.4: Example of an array representation with gaps**

**[SWS\_ComXf\_00014]** [ The COM Based Transformer shall calculate the 'COM Data Offset' by collecting all other transformer's `headerLength` information and add the start position inside the `ComIPdu`. ]([SRS\\_Xfrm\\_00201](#), [SRS\\_Xfrm\\_00202](#))

**[SWS\_ComXf\_00020]** [ The COM Based Transformer shall place the serialized data element into the data buffer at the position:

`ComBitPosition` - (8 \* 'COM Data Offset')

where `ComBitPosition` is taken from the `ComGroupSignal` definition in COM and 'COM Data Offset' is defined by [SWS\_ComXf\_00014]. ]([SRS\\_Xfrm\\_00201](#), [SRS\\_Xfrm\\_00202](#))

**[SWS\_ComXf\_00013]** [ To allow migration, the deserialization shall be able to accept larger array representations and ignore dataElements appended at the end of a previously known parameter list. ]([SRS\\_Xfrm\\_00201](#))

This means: data elements that were not defined in the interface specification used to generate or parameterize the deserialization code at the end of the serialized data will be ignored by the deserialization.

<b>Class TransformationTechnology</b>				
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Transformer			
<b>Note</b>	A TransformationTechnology is a transformer inside a transformer chain.  <b>Tags:</b> xml.namePlural=TRANSFORMATION-TECHNOLOGIES			
<b>Base</b>	ARObject, Identifiable, MultilanguageReferrable, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
bufferProperties	BufferProperties	1	aggr	Aggregation of the mandatory BufferProperties.
hasInternalState	Boolean	0..1	attr	This attribute defines whether the Transformer has an internal state or not.
needsOriginalData	Boolean	0..1	attr	Specifies whether this transformer gets access to the SWC's original data.
protocol	String	1	attr	Specifies the protocol that is implemented by this transformer.
transformationDescription	TransformationDescription	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the TransformerDescription.  <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild
transformerClass	TransformerClassEnum	1	attr	Specifies to which transformer class this transformer belongs.
version	String	1	attr	Version of the implemented protocol.

**Table 7.1: TransformationTechnology**

<b>Enumeration TransformerClassEnum</b>	
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Transformer
<b>Note</b>	Specifies the transformer class of a transformer.
<b>Literal</b>	<b>Description</b>
custom	The transformer is a custom transformer.  <b>Tags:</b> atp.EnumerationValue=0
safety	The transformer is a safety transformer.  <b>Tags:</b> atp.EnumerationValue=1
security	The transformer is a security transformer.  <b>Tags:</b> atp.EnumerationValue=2
serializer	The transformer is a serializing transformer.  <b>Tags:</b> atp.EnumerationValue=3

**Table 7.2: TransformerClassEnum**

<b>Class</b>	<b>BufferProperties</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Transformer			
<b>Note</b>	Configuration of the buffer properties the transformer needs to work.			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
bufferComputation	CompuScale	0..1	aggr	If the transformer changes the size of the data, the CompuScale can be used to specify a rule to derive the size of the output data based on the size of the input data.
headerLength	Integer	1	attr	Defines the length of the header (in bits) this transformer will add in front of the data.
inPlace	Boolean	1	attr	If set, the transformer uses the input buffer as output buffer.

**Table 7.3: BufferProperties**

<b>Class</b>	<b>TransformationDescription (abstract)</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Transformer			
<b>Note</b>	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.			
<b>Base</b>	ARObject, Describable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 7.4: TransformationDescription**

## 7.2 Error classification

### 7.2.1 Development Errors

**[SWS\_ComXf\_00028] Development Errors** [ The errors shown in table 7.5 are defined as development errors for the COM Based Transformer. ] ([SRS\\_BSW\\_00337](#))

Type of error	Related error code	Value
Error code if any other API service, except <code>GetVersionInfo</code> is called before the transformer module was initialized with <code>Init</code> or after a call to <code>DeInit</code>	<MIP>_E_UNINIT	0x01
Error code if an invalid configuration set was selected	<MIP>_E_INIT_FAILED	0x02
API service called with wrong parameter	<MIP>_E_PARAM	0x03
API service called with invalid pointer	<MIP>_E_PARAM_POINTER	0x04

**Table 7.5: Development Errors**

### **7.2.2 Runtime Errors**

No runtime errors are specified for the COM Based Transformer.

### **7.2.3 Transient Faults**

No transient faults are specified for the COM Based Transformer.

### **7.2.4 Production Errors**

No production errors are specified for the COM Based Transformer.

### **7.2.5 Extended Production Errors**

No extended production errors are specified for the COM Based Transformer.

## 8 API specification

### 8.1 Imported types

There are no imported types from other modules beyond those specified in [2].

In the Module Interlink Headers file which is imported by the COM Based Transformer, all `ImplementationDataTypes` known to the RTE are included. Using this mechanism, the COM Based Transformer knows all data types of data which shall be transformed.

### 8.2 Type definitions

[SWS\_ComXf\_00030] [

<b>Name:</b>	ComXf_ConfigType		
<b>Type:</b>	Structure		
<b>Element:</b>		implementation specific	–
<b>Description:</b>	This is the type of the data structure containing the initialization data for the transformer.		

**Table 8.1: ComXf\_ConfigType**

]([SRS\\_BSW\\_00404](#), [SRS\\_BSW\\_00441](#))

### 8.3 Function definitions

The COM Based transformer provides the specific interfaces generally required by [2].

[SWS\_ComXf\_00006] [ The COM Based Transformer shall only provide functions for transformers where the `TransformationTechnology` is referenced as the first reference in the list of ordered references `transformerChain` from a `DataTransformation` to a `TransformationTechnology`. ]([SRS\\_Xfrm\\_00011](#))

That means, only the first transformer in a transformer chain can be a COM Based Transformer because serializer transformer are in general only allowed to be the first transformer in a chain.

#### 8.3.1 ComXf\_<transformerId>

[SWS\_ComXf\_00007] [

<b>Service name:</b>	ComXf_<transformerId>
----------------------	-----------------------

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

**Table 8.2: ComXf\_<transformerId>**

|(SRS\_Xfrm\_00201)  
where

- `type` is data type of the data element after all data conversion activities of the RTE
- `paramtype` is derived from `type` according to the parameter passing rules defined by the *SRS BSW General* [5] (see [SRS\_BSW\_00484], [SRS\_BSW\_00485], and [SRS\_BSW\_00486]) and *SWS BSW General* [8] (see [SWS\_BSW\_00186] and [SWS\_BSW\_00187])
- `transformerId` is the name pattern for the transformer specified by the *General Specification on Transformers* [2] [SWS\_Xfrm\_00062].

The function specified in [SWS\_ComXf\_00007] exists for each transformed Sender/Receiver communication which uses the COM Based Transformer.

[SWS\_ComXf\_00008] [ The function specified in [SWS\_ComXf\_00007] shall exist for the first reference in the list of ordered references `transformerChain` from a `DataTransformation` to a `TransformationTechnology` if the `DataTransformation` is referenced by an `ISignalGroup` in the role `comBasedSignalGroupTransformation` where the `ISignalGroup` references a `SystemSignalGroup` which is referenced by `SenderReceiverToSignalGroupMapping`. ]  
(SRS\_Xfrm\_00201)

[SWS\_ComXf\_00009] [ The function specified in [SWS\_ComXf\_00007] shall serialize complex data elements of Sender/Receiver communication into a linear byte array representation using the COM Based Transformation. ](SRS\_Xfrm\_00201)

### 8.3.2 ComXf\_Inv\_<transformerId>

[SWS\_ComXf\_00010] [

<b>Service name:</b>	ComXf_Inv_<transformerId>	
<b>Syntax:</b>	uint8 ComXf_Inv_<transformerId>( const uint8* buffer, uint32 bufferLength, <type>* dataElement )	
<b>Service ID[hex]:</b>	0x04	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	buffer	Buffer allocated by the RTE, where the still serialized data are stored by the Rte
	bufferLength	Used length of the buffer
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	dataElement	Data element which is the result of the transformation and contains the deserialized data element
<b>Return value:</b>	uint8	0x00 (E_OK): Serialization successful 0x01 (E_NO_DATA): No data available which can be deserialized 0x81 (E_SER_GENERIC_ERROR): A generic error occurred
<b>Description:</b>	This function deserializes a Sender/Receiver communication using the deserialization of COM Based Transformer. It takes the uint8 array containing the serialized data as input and outputs the original data element which will be passed to the Rte.	

**Table 8.3: ComXf\_Inv\_<transformerId>**

](SRS\_Xfrm\_00201)  
where

- `type` is data type of the data element before all data conversion activities of the RTE
- `transformerId` is the name pattern for the transformer specified in [SWS\_Xfrm\_00062] ([2]).

The function specified in [SWS\_ComXf\_00010] exists for each transformed Sender/Receiver communication which uses the COM Based Transformation.

[SWS\_ComXf\_00011] [ The function specified in [SWS\_ComXf\_00010] shall exist for the first reference in the list of ordered references `transformerChain` from a `DataTransformation` to a `TransformationTechnology` if the `DataTransformation` is referenced by an `ISignalGroup` in the role `comBasedSignalGroupTransformation` where the `ISignalGroup` references a `SystemSignalGroup` which is referenced by `SenderReceiverToSignalGroupMapping`. ]  
(SRS\_Xfrm\_00201)

[SWS\_ComXf\_00035] [ If `ComXf_Inv_<transformerId>` specified in [SWS\_ComXf\_00010] is called with `buffer` equal to `NULL_PTR` and `bufferLength`

equal to 0, then the output buffer `buffer` shall not be changed and `ComXf_Inv_<transformerId>` shall return with `E_NO_DATA`. ]([SRS\\_Xfrm\\_00201](#))

**[SWS\_ComXf\_00012]** [ The function specified in [[SWS\\_ComXf\\_00010](#)] shall deserialize a linear byte array to primitive or complex data elements of Sender/Receiver communication using the COM Based Transformation. ]([SRS\\_Xfrm\\_00201](#))

### 8.3.3 ComXf\_Init

**[SWS\_ComXf\_00026]** [

<b>Service name:</b>	ComXf_Init	
<b>Syntax:</b>	<pre>void ComXf_Init( const ComXf_ConfigType* config )</pre>	
<b>Service ID[hex]:</b>	0x01	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	config	Pointer to the transformer's configuration data.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This service initializes the transformer for the further processing.	

**Table 8.4: ComXf\_Init**

]([SRS\\_BSW\\_00407](#), [SRS\\_BSW\\_00411](#))

### 8.3.4 ComXf\_DeInit

**[SWS\_ComXf\_00027]** [

<b>Service name:</b>	ComXf_DeInit	
<b>Syntax:</b>	<pre>void ComXf_DeInit( void )</pre>	
<b>Service ID[hex]:</b>	0x02	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This service deinitializes the transformer.	

**Table 8.5: ComXf\_DeInit**

]([SRS\\_BSW\\_00407](#), [SRS\\_BSW\\_00411](#))



### 8.3.5 ComXf\_GetVersionInfo

[SWS\_ComXf\_00024] [

<b>Service name:</b>	ComXf_GetVersionInfo	
<b>Syntax:</b>	void ComXf_GetVersionInfo( Std_VersionInfoType* VersionInfo )	
<b>Service ID[hex]:</b>	0x00	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	VersionInfo	Pointer to where to store the version information of this module.
<b>Return value:</b>	None	
<b>Description:</b>	This service returns the version information of the called transformer module.	

**Table 8.6: ComXf\_GetVersionInfo**

]([SRS\\_BSW\\_00407](#), [SRS\\_BSW\\_00411](#))

## 8.4 Callback notifications

COM Based Transformer has no callback notifications.

## 8.5 Scheduled functions

COM Based Transformer has no scheduled functions.

## 8.6 Expected interfaces

COM Based Transformer has no expected interfaces.

## 9 Sequence diagrams

There are no sequence diagrams applicable to COM Based Transformer.

## 10 Configuration specification

**[SWS\_ComXf\_00031]** [ The COM Based Transformer is configured based on the COM module configuration [4]. ]([SRS\\_Xfm\\_00202](#))

Still, there is an EcuC necessary to map the implementation of the transformer. The EcuC defined in [2] shall be used.

**[SWS\_ComXf\_00033]** [ The vendor specific module definition of the COM Based Transformer - based on the Xfm configuration [2] - may be extended by the vendor to support the close interaction with the Com module [4]. ]([SRS\\_Xfm\\_00202](#))

**[SWS\_ComXf\_00034]** [ The COM Based Transformer shall be configured to be `postBuild` when the configuration of the Com module [4] is `postBuild` for the respective `ComSignalGroups`. ]([SRS\\_Xfm\\_00202](#))

**[SWS\_ComXf\_00025]** [ The `apiServicePrefix` of the COM Based Transformer's EcuC shall be set to `ComXf`. ]([SRS\\_BSW\\_00159](#), [SRS\\_Xfm\\_00202](#))

## A Referenced Meta Classes

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

<b>Class</b>	<b>BswImplementation</b>			
<b>Package</b>	M2::AUTOSARTemplates::BswModuleTemplate::BswImplementation			
<b>Note</b>	<p>Contains the implementation specific information in addition to the generic specification (BswModuleDescription and BswBehavior). It is possible to have several different BswImplementations referring to the same BswBehavior.</p> <p><b>Tags:</b> atp.recommendedPackage=BswImplementations</p>			
<b>Base</b>	ARElement, ARObject, CollectableElement, Identifiable, <a href="#">Implementation</a> , MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
arReleaseVersion	RevisionLabelString	1	attr	Version of the AUTOSAR Release on which this implementation is based. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.
behavior	BswInternalBehavior	1	ref	<p>The behavior of this implementation.</p> <p>This relation is made as an association because</p> <ul style="list-style-type: none"> <li>• it follows the pattern of the SWCT</li> <li>• since ARElement cannot be splitted, but we want supply the implementation later, the BswImplementation is not aggregated in BswBehavior</li> </ul>
preconfiguredConfiguration	EcucModuleConfigurationValues	*	ref	<p>Reference to the set of preconfigured (i.e. fixed) configuration values for this BswImplementation.</p> <p>If the BswImplementation represents a cluster of several modules, more than one EcucModuleConfigurationValues element can be referred (at most one per module), otherwise at most one such element can be referred.</p> <p><b>Tags:</b> xml.roleWrapperElement=true</p>
recommendedConfiguration	EcucModuleConfigurationValues	*	ref	Reference to one or more sets of recommended configuration values for this module or module cluster.

<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
vendorApilnfix	Identifier	0..1	attr	<p>In driver modules which can be instantiated several times on a single ECU, SRS_BSW_00347 requires that the names of files, APIs, published parameters and memory allocation keywords are extended by the vendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific API name is generated as follows: &lt;ModuleName&gt;_&lt;vendorId&gt;_&lt;vendorApilnfix&gt;_&lt;API name from SWS&gt;.</p> <p>E.g. assuming that the vendorId of the implementer is 123 and the implementer chose a vendorApilnfix of "v11r456" an API name Can_Write defined in the SWS will translate to Can_123_v11r456_Write.</p> <p>This attribute is mandatory for all modules with upper multiplicity &gt; 1. It shall not be used for modules with upper multiplicity =1.</p> <p>See also SWS_BSW_00102.</p>
vendorSpecificModuleDef	<a href="#">EcucModuleDef</a>	*	ref	<p>Reference to</p> <ul style="list-style-type: none"> <li>the vendor specific EcucModuleDef used in this BswImplementation if it represents a single module</li> <li>several EcucModuleDefs used in this BswImplementation if it represents a cluster of modules</li> <li>one or no EcucModuleDefs used in this BswImplementation if it represents a library</li> </ul> <p><b>Tags:</b> xml.roleWrapperElement=true</p>

**Table A.1: BswImplementation**

<b>Class</b>	<b>DataTransformation</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Transformer			
<b>Note</b>	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
<b>Base</b>	ARObject, Identifiable, MultilanguageReferrable, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataTransformationKind	DataTransformationKindEnum	0..1	attr	This attribute controls the kind of DataTransformation to be applied.
executeDespiteDataUnavailability	Boolean	1	attr	Specifies whether the transformer chain is executed even if no input data are available.

Attribute	Type	Mul.	Kind	Note
transformerChain (ordered)	<a href="#">Transformation Technology</a>	1..*	ref	This attribute represents the definition of a chain of transformers that are supposed to be executed according to the order of being referenced from DataTransformation.

**Table A.2: DataTransformation**

Class	EcucModuleDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used as the top-level element for configuration definition for Software Modules, including BSW and RTE as well as ECU Infrastructure.  <b>Tags:</b> atp.recommendedPackage=EcucModuleDefs			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpDefinition, Collectable Element, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	Type	Mul.	Kind	Note
apiServicePrefix	CIdentifier	0..1	attr	For CDD modules this attribute holds the apiServicePrefix.  The shortName of the module definition of a Complex Driver is always "Cdd". Therefore for CDD modules the module apiServicePrefix is described with this attribute.
container	EcucContainerDef	1..*	aggr	Aggregates the top-level container definitions of this specific module definition.  <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=shortName xml.sequenceOffset=11
postBuildVariantSupport	Boolean	0..1	attr	Indicates if a module supports different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
refinedModuleDef	<a href="#">EcucModuleDef</a>	0..1	ref	Optional reference from the Vendor Specific Module Definition to the Standardized Module Definition it refines. In case this EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION this reference shall not be provided. In case this EcucModuleDef has the category VENDOR_SPECIFIC_MODULE_DEFINITION this reference is mandatory.  <b>Stereotypes:</b> atpUriDef
supportedConfigVariant	EcucConfigurationVariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION then this attribute is mandatory.

Attribute	Type	Mul.	Kind	Note
-----------	------	------	------	------

**Table A.3: EcucModuleDef**

Class	ISignalGroup			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	<p>SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers.</p> <p>An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group.</p> <p>Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)</p> <p><b>Tags:</b> atp.recommendedPackage=ISignalGroup</p>			
<b>Base</b>	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mul.	Kind	Note
comBasedSignalGroupTransformation	DataTransformation	0..1	ref	<p>Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=comBasedSignalGroupTransformation, variationPoint.shortLabel.vh.latestBindingTime=codeGenerationTime</p>
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together.
systemSignalGroup	SystemSignalGroup	1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup.
transformationSignalProps	TransformationSignalProps	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationSignalProps class. The transformer configuration properties that are common for all ISignalGroups are described in the TransformationTechnology class.

**Table A.4: ISignalGroup**

<b>Class</b>	<b>ISignalPdu</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	<p>Represents the IPdus handled by Com. The ISignalPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer.</p> <p>A maximum of one dynamic length signal per IPdu is allowed.</p> <p><b>Tags:</b> atp.recommendedPackage=Pdus</p>			
<b>Base</b>	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, Multilanguage Referrable, PackageableElement, Pdu, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
iPduTiming Specification	IPduTiming	0..1	aggr	<p>Timing specification for Com IPdus (Transmission Modes). This information is mandatory for the sender in a System Extract. This information may be omitted on receivers in a System Extract.</p> <p>atpVariation: The timing of a Pdu can vary.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild</p>
iSignalToPduMapping	<a href="#">ISignalToIPduMapping</a>	*	aggr	<p>Definition of SignalToIPduMappings included in the SignalIPdu.</p> <p>atpVariation: The content of a PDU can be variable.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild</p>
pduCounter	SignalIPduCounter	0..1	aggr	<p>An included Pdu counter is used to ensure that a sequence of Pdus is maintained.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime</p>
pduReplication	SignalIPduReplication	0..1	aggr	<p>Pdu Replication is a form of redundancy where the data content of one ISignalIPdu (source) is transmitted inside a set of replica ISignalIPdus. These ISignalIPdus (copies) have different Pdu IDs, identical PduCounters, identical data content and are transmitted with the same frequency.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime</p>
unusedBitPattern	Integer	1	attr	<p>AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.</p>

**Table A.5: ISignalPdu**



<b>Class</b>	<b>ISignalToIPduMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
<b>Base</b>	ARObject, Identifiable, MultilanguageReferrable, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
iSignal	ISignal	0..1	ref	<p>Reference to a ISignal that is mapped into the ISignalIPdu.</p> <p>Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive.</p>
iSignalGroup	<a href="#">ISignalGroup</a>	0..1	ref	<p>Reference to an ISignalGroup that is mapped into the SignalIPdu. If an ISignalToIPduMapping for an ISignalGroup is defined, only the UpdateIndicationBitPosition and the transferProperty is relevant. The startPosition and the packingByteOrder shall be ignored.</p> <p>Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive.</p>
packingByteOrder	ByteOrderEnum	0..1	attr	<p>This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description).</p> <p>For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.</p>

<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
startPosition	Integer	0..1	attr	<p>This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.</p> <p>Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array.</p> <p>If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.</p>
transferProperty	TransferPropertyEnum	0..1	attr	<p>The triggered or triggeredOnChange, triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition transferProperty causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU.</p> <p>The immediate transmission of the IPdu is caused even if only one Signal of an IPdu has the transferProperty triggered or triggeredWithoutRepetition or triggeredOnChange or triggeredOnChangeWithoutRepetition and all other Signals have the transferProperty pending.</p> <p>Also for ISignals of an ISignalGroup (GroupSignals) this attribute is relevant and shall be evaluated:</p> <ul style="list-style-type: none"> <li>• If none of the ISignals belonging to the ISignalGroup have a transferProperty defined the transferProperty of the ISignalToPduMapping referring to the ISignalGroup is considered.</li> <li>• If at least one of the ISignals belonging to the ISignalGroup has a transferProperty defined all other ISignals belonging to the same ISignalGroup shall have a transferProperty defined as well. All of the transferProperties of the GroupSignals are considered.</li> </ul>

<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
updateIndicationBitPosition	Integer	0..1	attr	<p>The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored.</p> <p>Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByteLast. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change.</p> <p>This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.</p>

**Table A.6: ISignalToIPduMapping**

<b>Class</b>	<b>Implementation (abstract)</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::Implementation			
<b>Note</b>	Description of an implementation a single software component or module.			
<b>Base</b>	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
buildActionManifest	BuildActionManifest	0..1	ref	<p>A manifest specifying the intended build actions for the software delivered with this implementation.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=codeGenerationTime</p>
codeDescriptor	Code	1..*	aggr	Specifies the provided implementation code.
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released

<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
generatedArtifact	DependencyOnArtifact	*	aggr	<p>Relates to an artifact that will be generated during the integration of this Implementation by an associated generator tool. Note that this is an optional information since it might not always be in the scope of a single module or component to provide this information.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime</p>
hwElement	HwElement	*	ref	<p>The hardware elements (e.g. the processor) required for this implementation.</p>
linker	Linker	*	aggr	<p>Specifies the linker for which this implementation has been released.</p>
mcSupport	McSupportData	0..1	aggr	<p>The measurement &amp; calibration support data belonging to this implementation. The aggregation is «atpSplitable» because in case of an already existing BSW Implementation model, this description will be added later in the process, namely at code generation time.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=mcSupport</p>
programmingLanguage	ProgrammingLanguageEnum	1	attr	<p>Programming language the implementation was created in.</p>
requiredArtifact	DependencyOnArtifact	*	aggr	<p>Specifies that this Implementation depends on the existence of another artifact (e.g. a library). This aggregation of DependencyOnArtifact is subject to variability with the purpose to support variability in the implementations. Different algorithms in the implementation might cause different dependencies, e.g. the number of used libraries.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime</p>
requiredGeneratorTool	DependencyOnArtifact	*	aggr	<p>Relates this Implementation to a generator tool in order to generate additional artifacts during integration.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime</p>
resourceConsumption	ResourceConsumption	1	aggr	<p>All static and dynamic resources for each implementation are described within the ResourceConsumption class.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=shortName</p>
swVersion	RevisionLabelString	1	attr	<p>Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.</p>

<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
swcBswMapping	SwcBswMapping	0..1	ref	This allows a mapping between an SWC and a BSW behavior to be attached to an implementation description (for AUTOSAR Service, ECU Abstraction and Complex Driver Components). It is up to the methodology to define whether this reference has to be set for the Swc- or BswImplementation or for both.
usedCodeGenerator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

**Table A.7: Implementation**

<b>Class</b>	<b>ImplementationDataType</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
<b>Note</b>	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code.  <b>Tags:</b> atp.recommendedPackage=ImplementationDataTypes			
<b>Base</b>	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Specifies an element of an array, struct, or union data type.  The aggregation of ImplementationDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure.  <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the ImplementationDataType.  <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=shortName
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

**Table A.8: ImplementationDataType**

<b>Class</b>	<b>PortPrototype (abstract)</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Components			
<b>Note</b>	Base class for the ports of an AUTOSAR software component.  The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
<b>Base</b>	ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, Multilanguage Referrable, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPortAnnotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServerAnnotation	IoHwAbstractionServerAnnotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePortAnnotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPortAnnotation	NvDataPortAnnotation	*	aggr	Annotations on this non volatile data port.
parameterPortAnnotation	ParameterPortAnnotation	*	aggr	Annotations on this parameter port.
senderReceiverAnnotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPortAnnotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

**Table A.9: PortPrototype**

<b>Class</b>	<b>SenderReceiverInterface</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
<b>Note</b>	A sender/receiver interface declares a number of data elements to be sent and received.  <b>Tags:</b> atp.recommendedPackage=PortInterfaces			
<b>Base</b>	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	VariableDataPrototype	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement

**Table A.10: SenderReceiverInterface**

<b>Class</b>	<b>SenderReceiverToSignalGroupMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
<b>Note</b>	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
<b>Base</b>	ARObject, DataMapping			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	VariableDataPrototype	1	iref	Reference to a data element with a composite datatype which is mapped to a signal group.
signalGroup	<a href="#">SystemSignalGroup</a>	1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type
typeMapping	SenderRecCompositeTypeMapping	1	aggr	The CompositeTypeMapping maps the the ApplicationArrayElements and ApplicationRecordElements to Signals of the SignalGroup.

**Table A.11: SenderReceiverToSignalGroupMapping**

<b>Class</b>	<b>SystemSignalGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	<p>A signal group refers to a set of signals that must always be kept together. A signal group is used to guarantee the atomic transfer of AUTOSAR composite data types.</p> <p>The SystemSignalGroup defines a signal grouping on VFB level. On cluster level the Signal grouping is described by the ISignalGroup element.</p> <p><b>Tags:</b> atp.recommendedPackage=SystemSignalGroups</p>			
<b>Base</b>	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that must always be kept together.
transformingSystemSignal	SystemSignal	0..1	ref	Optional reference to the SystemSignal which shall contain the transformed (linear) data.

**Table A.12: SystemSignalGroup**

## B Used ECU Configuration

To ease the readability the relevant parts of the Com module configuration [4] are duplicated here (for information only).

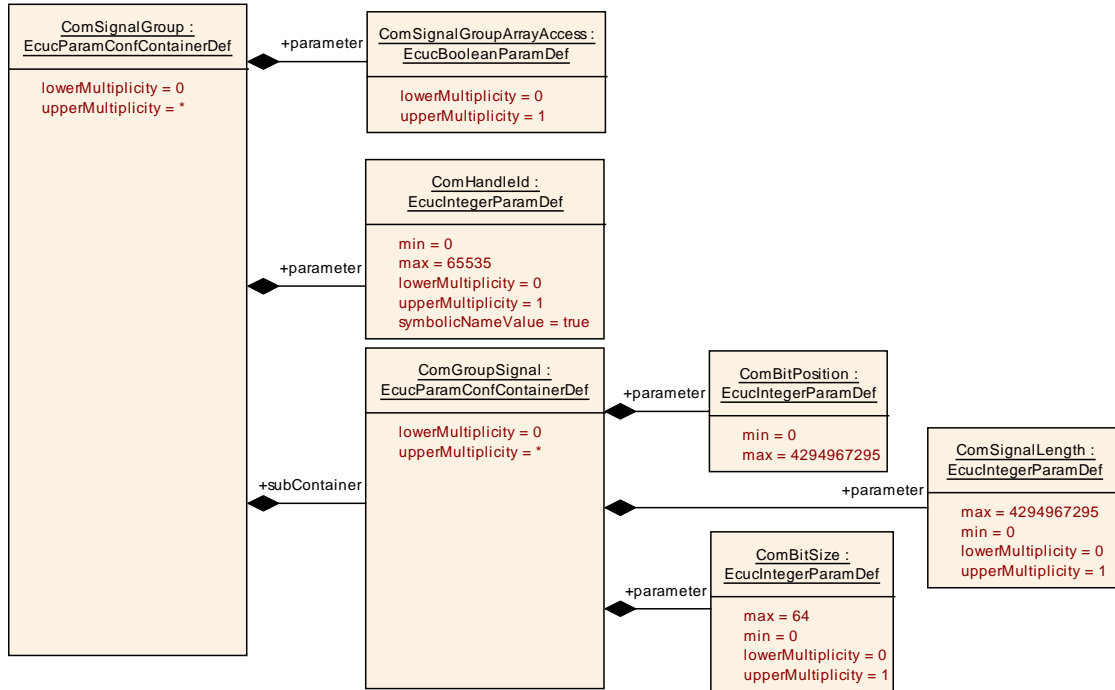


Figure B.1: AR\_EcucDef\_ComSignalGroupArrayAccess

<b>SWS Item</b>	[ECUC_Com_00345]		
<b>Container Name</b>	ComSignalGroup		
<b>Description</b>	Contains the configuration parameters of the AUTOSAR COM module's signal groups.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>Name</b>	ComDataInvalidAction [ECUC_Com_00314]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	This parameter defines the action performed upon reception of an invalid signal. Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NOTIFY		
	REPLACE		Literal for DataInvalidAction



<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComErrorNotification [ECUC_Com_00499]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Only valid on sender side: Name of Com_CbkTxErr callback function to be called. If this parameter is omitted no error notification shall take place.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComFirstTimeout [ECUC_Com_00183]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. 3600]		

<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComHandleId [ECUC_Com_00165]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	<p>The numerical value used as the ID.</p> <p>This ID identifies signals and signal groups in the COM APIs using Com_SignalIdType or Com_SignalGroupIdType parameter respectively.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	ComInitialValueOnly [ECUC_Com_00811]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	<p>This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>	false		

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComInvalidNotification [ECUC_Com_00315]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComNotification [ECUC_Com_00498]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	On sender side: Name of Com_CbkTxAck callback function to be called. On receiver side: Name of Com_CbkRxAck callback function to be called.  If this parameter is omitted no notification shall take place.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComRxDataTimeoutAction [ECUC_Com_00412]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInitValue	
	SUBSTITUTE	signals shall be replaced by their ComTimeoutSubstitutionValue	
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComSignalGroupArrayAccess [ECUC_Com_10003]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Defines whether the uint8-array based access shall be used for this ComSignalGroup.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>			

<b>Name</b>	ComTimeout [ECUC_Com_00263]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. 3600]		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Scope / Dependency</b>	scope: local
---------------------------	--------------

<b>Name</b>	ComTimeoutNotification [ECUC_Com_00552]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComTransferProperty [ECUC_Com_00232]	
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>	
<b>Description</b>	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.
	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.
	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.

<b>Post-Build Variant Multiplicity</b>	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.	
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComUpdateBitPosition [ECUC_Com_00257]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComSystemTemplateSignalGroupRef [ECUC_Com_00001]		
<b>Parent Container</b>	<a href="#">ComSignalGroup</a>		
<b>Description</b>	Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">ComGroupSignal</a>	0..*	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.

<b>SWS Item</b>	[ECUC_Com_00520]		
<b>Container Name</b>	ComGroupSignal		
<b>Description</b>	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>Name</b>	ComBitPosition [ECUC_Com_00259]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default Value</b>			



<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComBitSize [ECUC_Com_00158]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 64		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComHandleId [ECUC_Com_00165]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	The numerical value used as the ID.  This ID identifies signals and signal groups in the COM APIs using Com_SignalIdType or Com_SignalGroupIdType parameter respectively.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	ComSignalDataInvalidValue [ECUC_Com_00391]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	<p>Defines the data invalid value of the signal.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

<b>Name</b>	ComSignalEndianness [ECUC_Com_00157]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	Defines the endianness of the signal's network representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN		
	LITTLE_ENDIAN		
	OPAQUE		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComSignalInitValue [ECUC_Com_00170]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	<p>Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default Value</b>	0		
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

<b>Name</b>	ComSignalLength [ECUC_Com_00437]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	<p>Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.</p> <p>The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComSignalType [ECUC_Com_00127]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BOOLEAN		
	FLOAT32		

<b>Post-Build Variant Value</b>	FLOAT64		
	SINT16		
	SINT32		
	SINT64		
	SINT8		
	UINT16		
	UINT32		
	UINT64		
	UINT8		
	UINT8_DYN		
	UINT8_N	false	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComTimeoutSubstitutionValue [ECUC_Com_10006]
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>
<b>Description</b>	<p>The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00.</p> <p>In case ofUINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification.</p> <p>In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>
<b>Multiplicity</b>	0..1
<b>Type</b>	EcucStringParamDef
<b>Default Value</b>	

<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComTransferProperty [ECUC_Com_00560]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	Optionally defines whether this group signal shall contribute to the TRIGGERED_ON_CHANGE transfer property of the signal group. If at least one group signal of a signal group has the "ComTransferProperty" configured all other group signals of that signal group shall have the attribute configured as well.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	PENDING	A change of the value of this group signal shall not be considered in the evaluation of the signal groups ComTransferProperty.	
	TRIGGERED_ON_CHANGE	A change of the value of this group signal shall be considered in the evaluation of the signal groups ComTransferProperty.	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComSystemTemplateSystemSignalRef [ECUC_Com_00002]		
<b>Parent Container</b>	<a href="#">ComGroupSignal</a>		
<b>Description</b>	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComFilter	0..1	This container contains the configuration parameters of the AUTOSAR COM module's Filters.  Note: On sender side the container is used to specify the transmission mode conditions.

<b>SWS Item</b>	[ECUC_Com_00340]		
<b>Container Name</b>	ComIPdu		
<b>Description</b>	Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>Name</b>	ComIPduCallout [ECUC_Com_00387]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	This parameter defines the existence and the name of a callout function for the corresponding I-PDU. If this parameter is omitted no I-PDU callout shall take place for the corresponding I-PDU.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduCancellationSupport [ECUC_Com_00709]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	<p>Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF-modul supports cancellation of transmit requests.</p> <p>Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>	false		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	–	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	–	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: This parameter shall not be set to true if ComCancellationSupport is set to false		

<b>Name</b>	ComIPduDirection [ECUC_Com_00493]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	RECEIVE		
	SEND		
<b>Post-Build Variant Value</b>	false		



<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC_Com_00496		

<b>Name</b>	ComIPduHandleId [ECUC_Com_00175]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	The numerical value used as the ID of this I-PDU. The ComIPduHandleId is required by the API calls Com_RxIndication, Com_TpRxIndication, Com_StartOfReception and Com_CopyRxData to receive I-PDUs from the PduR (ComIP-duDirection: Receive), as well as the PduId passed to an Rx-I-PDU-callout. For Tx-I-PDUs (ComIPduDirection: Send), this handle Id is used for the APIs calls Com_TxConfirmation, Com_TriggerTransmit, Com_TriggerIPDUSend or Com_TriggerIPDUSendWithMetaData, Com_CopyTxData and Com_TpTxConfirmation to transmit respectively confirm transmissions of I-PDUs, as well as the PduId passed to the Tx-I-PDU-callout configured with ComIPduCallout and/or ComIPduTriggerTransmitCallout.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	ComIPduSignalProcessing [ECUC_Com_00119]	
<b>Parent Container</b>	<a href="#">ComIPdu</a>	
<b>Description</b>	For the definition of the two modes Immediate and Deferred.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	DEFERRED	signal indication / confirmations are deferred for example to a cyclic task
	IMMEDIATE	the signal indications / confirmations are performed in Com_RxIndication/ Com_TxConfirmation

<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduTriggerTransmitCallout [ECUC_Com_00765]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduType [ECUC_Com_00761]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NORMAL	sent or received via normal L-PDU	
	TP	sent or received via TP	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduGroupRef [ECUC_Com_00206]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	Reference to the I-PDU groups this I-PDU belongs to.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to ComIPduGroup		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduSignalGroupRef [ECUC_Com_00519]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	References to all signal groups contained in this I-Pdu		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to ComSignalGroup		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComIPduSignalRef [ECUC_Com_00518]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	References to all signals contained in this I-PDU.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to ComSignal		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComPduldRef [ECUC_Com_00711]		
<b>Parent Container</b>	<a href="#">ComIPdu</a>		
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to Pdu		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComIPduCounter	0..1	This optional container contains the configuration parameters of PDU Counter.
ComIPduReplication	0..1	This optional container contains the information needed for each I-PDU replicated.
<a href="#">ComTxIPdu</a>	0..1	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.

<b>SWS Item</b>	[ECUC_Com_00496]
<b>Container Name</b>	ComTxIPdu
<b>Description</b>	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.
<b>Configuration Parameters</b>	

<b>Name</b>	ComMetaDataDefault [ECUC_Com_10005]		
<b>Parent Container</b>	<a href="#">ComTxIPdu</a>		
<b>Description</b>	In case an I-PDU refers to a globally configured MetaDataType and no explicit meta data is given for a send request (e.g. by using Com_TriggerIPDUSendWithMetaData ), the AUTOSAR COM module uses this configured default meta data for sending. The configured string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address).		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU dependency: This parameter shall only be used in case ComMetaDataSupport is enabled. The length of the default meta data shall match the globally configured MetaDataLength of this I-PDU.		

<b>Name</b>	ComMinimumDelayTime [ECUC_Com_00181]		
<b>Parent Container</b>	<a href="#">ComTxIPdu</a>		
<b>Description</b>	Defines the Minimum Delay Time (MDT) between successive transmissions of this I-PDU in seconds. The MDT is independent of the possible different transmission modes. There is only one minimum delay time parameter for one I-PDU. The minimum delay timer is not reset by changing the transmission mode. Hence, it is not allowed to violate the minimum delay time by transmission mode changes. It is not possible to monitor the minimum delay time for I-PDUs that are requested using the Com_TriggerTransmit API.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. 3600]		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComTxIPduClearUpdateBit [ECUC_Com_00576]		
<b>Parent Container</b>	<a href="#">ComTxIPdu</a>		
<b>Description</b>	Defines when the update-bits of signals or signal groups, contained in this I-PDU, will be cleared.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	Confirmation	The update-bits are cleared when the transmission of the I-PDU was confirmed. In case of Direct/N-Times transmission mode the update bits will be cleared with respect to the confirmation behaviour of SWS_Com_00305.	
	Transmit	The update-bits are cleared directly after the invocation of PduR_ComTransmit.	
	TriggerTransmit	The update-bits are cleared after the I-PDU was fetched via Com_TriggerTransmit.	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	ComTxIPduUnusedAreasDefault [ECUC_Com_00017]		
<b>Parent Container</b>	<a href="#">ComTxIPdu</a>		
<b>Description</b>	The AUTOSAR COM module fills not used areas of an I-PDU with this byte pattern. This attribute is mandatory to avoid undefined behaviour. This byte-pattern will be repeated throughout the I-PDU before any init-values or update-bits were set.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
ComTxModeFalse	0..1	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to false.
ComTxModeTrue	0..1	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to true.

## C History of Constraints and Specification Items

### C.1 Constraint History of this Document according to AUTOSAR R4.2.1

Initial document release.

### C.2 Constraint History of this Document according to AUTOSAR R4.2.2

#### C.2.1 Added Traceables in 4.2.2

[\[SWS\\_ComXf\\_00032\]](#)

#### C.2.2 Changed Traceables in 4.2.2

[\[SWS\\_ComXf\\_00028\]](#)

#### C.2.3 Deleted Traceables in 4.2.2

none

### C.3 Constraint History of this Document according to AUTOSAR R4.3.0

#### C.3.1 Added Traceables in 4.3.0

[\[SWS\\_ComXf\\_00033\]](#) [\[SWS\\_ComXf\\_00034\]](#) [\[SWS\\_ComXf\\_00035\]](#)

#### C.3.2 Changed Traceables in 4.3.0

[\[SWS\\_ComXf\\_00001\]](#)

[\[SWS\\_ComXf\\_00004\]](#)

[\[SWS\\_ComXf\\_00006\]](#)

[\[SWS\\_ComXf\\_00007\]](#)

[\[SWS\\_ComXf\\_00008\]](#)

[\[SWS\\_ComXf\\_00010\]](#)

[\[SWS\\_ComXf\\_00011\]](#)



### **C.3.3 Deleted Traceables in 4.3.0**

none

## **C.4 Constraint History of this Document according to AUTOSAR R4.3.1**

### **C.4.1 Added Traceables in 4.3.1**

[\[SWS\\_ComXf\\_00036\]](#)

### **C.4.2 Changed Traceables in 4.3.1**

[\[SWS\\_ComXf\\_00007\]](#) [\[SWS\\_ComXf\\_00023\]](#)

### **C.4.3 Deleted Traceables in 4.3.1**

none