Document Title	Specification of COM Based Transformer
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	662

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R22-11

Document Change History

Date	Release	Changed by	Description
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes
2021-11-25	R21-11	AUTOSAR Release Management	• Updated buffer reservation in transformer chain
2020-11-30	R20-11	AUTOSAR Release Management	• No content changes
2019-11-28	R19-11	AUTOSAR Release Management	• editorial • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Updated buffer handling • Removed include file structure
2017-12-08	4.3.1	AUTOSAR Release Management	• Updated handling of gaps in the array representation of a signal group. • Clarification on parameter passing.
2016-11-30	4.3.0	AUTOSAR Release Management	• Updated include file structure figure. • Clarification on postBuild configuration in chapter 10. • Added support for unqueued communication when no data is available in [SWS_ComXf_00035].

2015-07-31	4.2.2	AUTOSAR Release Management	• Exclude support for external trigger communication [SWS_ComXf_00032]
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

Contents

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

References

[1] Glossary
AUTOSAR_TR_Glossary

[2] General Specification of 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)*

If the use-case occurs that a single primitive data element shall be handled by the COM Based Transformer (because there shall be a E2E protection performed after the serialization) the data element shall be wrapped in a structure. The structure would then just contain one entry, the signal group on COM level would also have the E2E parts included.

[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 exists 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].

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_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_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] [SWS_ComXf_00037]
[SRS_Xfrm_00202]	The COM Based Transformer shall take its configuration from the COM module	[SWS_ComXf_00005] [SWS_ComXf_00020] [SWS_ComXf_00025] [SWS_ComXf_00031] [SWS_ComXf_00033] [SWS_ComXf_00034] [SWS_ComXf_00036] [SWS_ComXf_00037]

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.

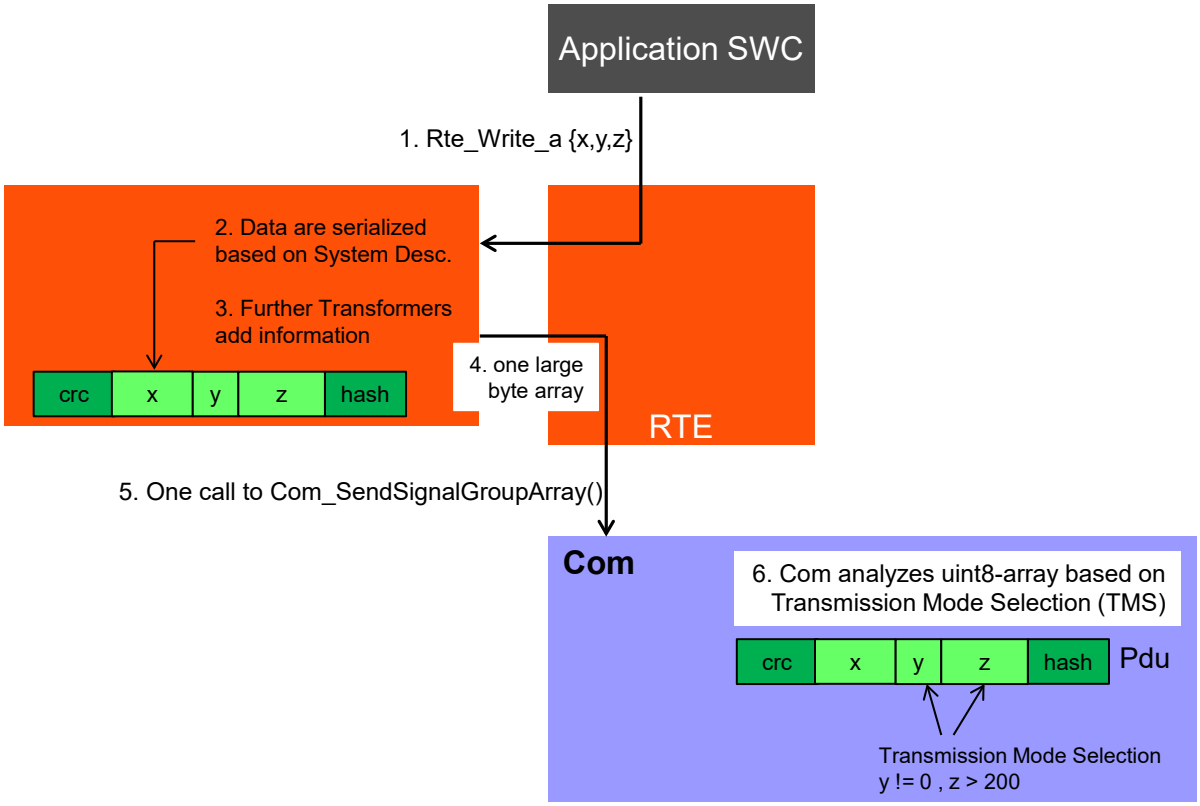

Figure 7.1: Overview of COM Based Transformer

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.

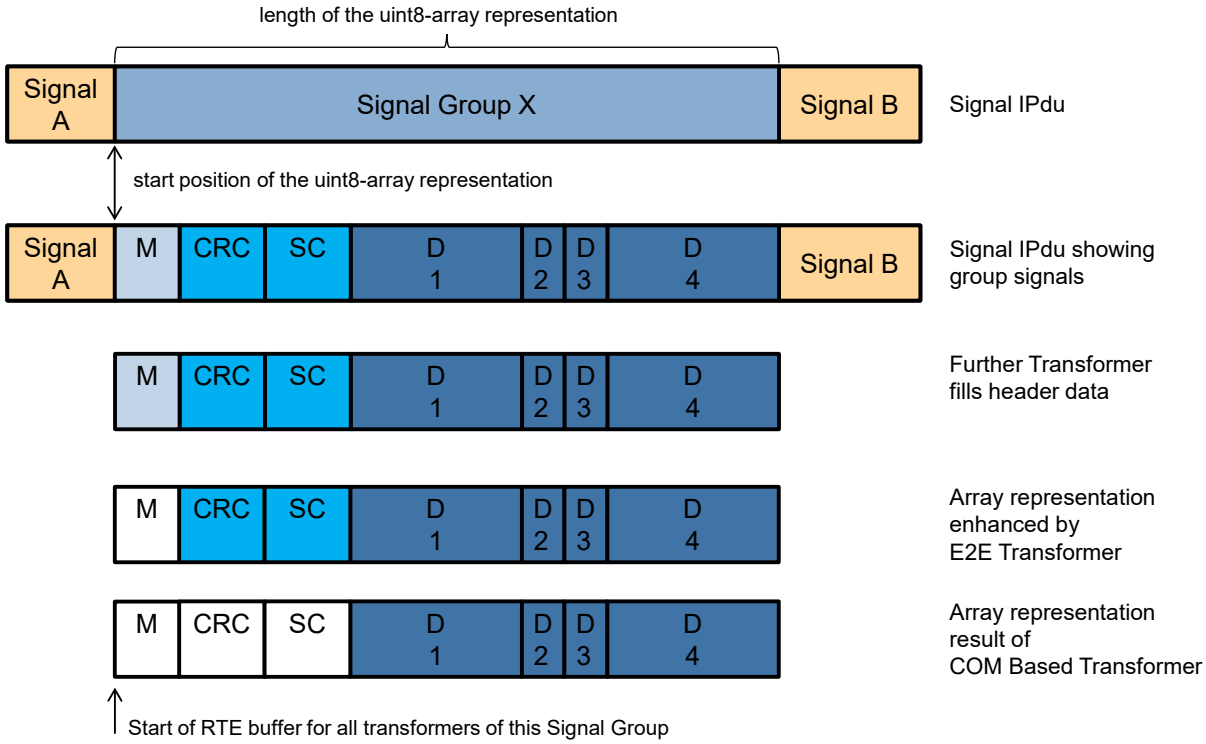

Figure 7.2: IPdu and signal layout

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).

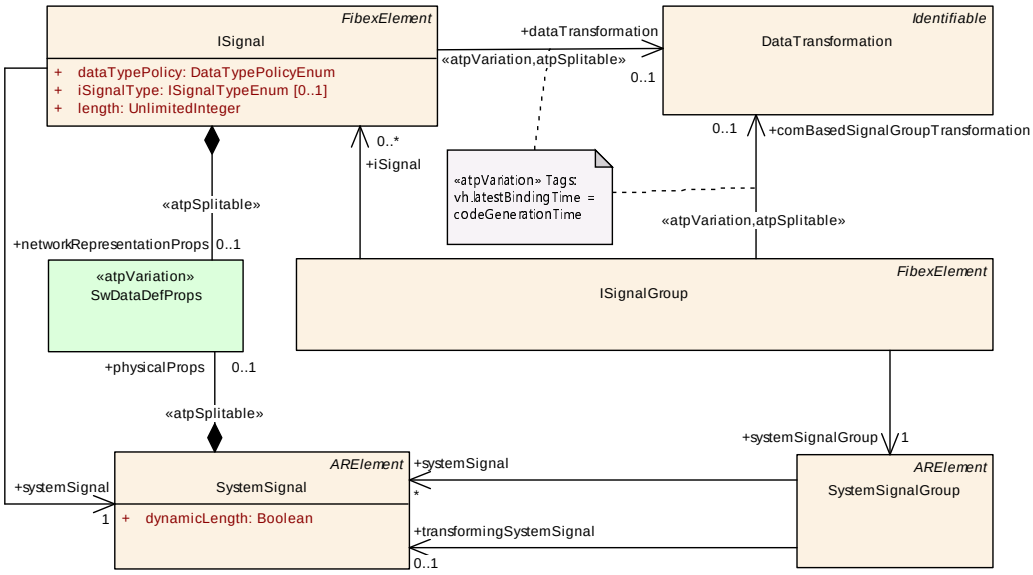

Figure 7.3: System Template Transformed communication

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 'M', '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 'M' signal will neither be handled by the E2E transformer nor the COM Based Transformer. For 'M' a further transformer is responsible.

The RTE will provide the data buffer beginning at the 'Start of RTE buffer for all transformers of this Signal Group' location to the COM Based Transformer, the E2E Transformer, and any further transformer defined.

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 topmost 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 `COM-Based`
- 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 endianess 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).

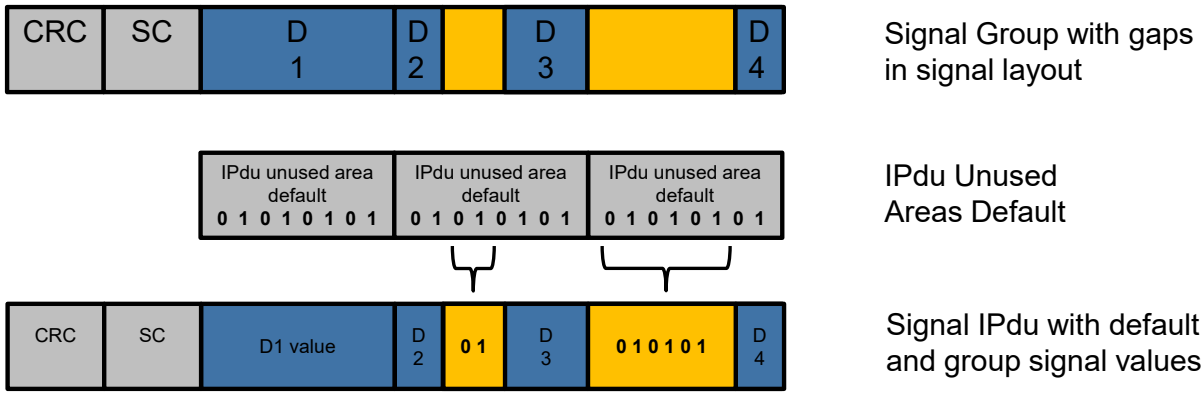

Figure 7.4: Example of an array representation with gaps

[SWS_ComXf_00037] Buffer reservation for further transformers ⌈The COM Based Transformer shall consider the header and/or trailer `ISignal`s defined in the `ISignalGroup` which are processed by further transformers.⌋*(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 bit position according to the configuration of the `ISignalGroup` in Com.⌋*(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.

Class	**TransformationTechnology**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A TransformationTechnology is a transformer inside a transformer chain. **Tags:**xml.namePlural=TRANSFORMATION-TECHNOLOGIES			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.transformationTechnology			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bufferProperties	BufferProperties	1	aggr	Aggregation of the mandatory BufferProperties.
hasInternal State	Boolean	0..1	attr	This attribute defines whether the Transformer has an internal state or not.
needsOriginal Data	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.
transformation Description	Transformation Description	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the Transformer Description. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=transformationDescription, transformation Description.variationPoint.shortLabel vh.latestBindingTime=postBuild
transformer Class	TransformerClassEnum	1	attr	Specifies to which transformer class this transformer belongs.
version	String	1	attr	Version of the implemented protocol.

Table 7.1: TransformationTechnology

Enumeration	**TransformerClassEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Specifies the transformer class of a transformer.
Aggregated by	TransformationTechnology.transformerClass
Literal	***Description***
custom	The transformer is a custom transformer. **Tags:**atp.EnumerationLiteralIndex=0
safety	The transformer is a safety transformer. **Tags:**atp.EnumerationLiteralIndex=1
security	The transformer is a security transformer. **Tags:**atp.EnumerationLiteralIndex=2
serializer	The transformer is a serializing transformer. **Tags:**atp.EnumerationLiteralIndex=3

Table 7.2: TransformerClassEnum

Class	**BufferProperties**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	*ARObject*			
Aggregated by	TransformationTechnology.bufferProperties			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
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

Class	***TransformationDescription*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.			
Base	*ARObject*, *Describable*			
Subclasses	EndToEndTransformationDescription, SOMEIPTransformationDescription, UserDefinedTransformation Description			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.4: TransformationDescription

7.2 Error classification

Section 7.2 "Error Handling" of the document "General Specification of Basic Software Modules" [8] describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.2.1 Development Errors

[SWS_ComXf_00028] ⌈

Type of error	*Related error code*	*Error value*
Error code if any other API service, except Get VersionInfo, is called before the transformer module was initialized with Init or after a call to De Init.	<MIP>_E_UNINIT	0x01

▽

△

Type of error	*Related error code*	*Error value*
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

⌋(*SRS_BSW_00337*)

7.2.2 Runtime Errors

There are no runtime errors.

7.2.3 Transient Faults

There are no transient faults.

7.2.4 Production Errors

There are no production errors.

7.2.5 Extended Production Errors

There are no extended production errors.

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 `ImplementationDataType`s 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] ⌈

Name	ComXf_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–
Description	This is the type of the data structure containing the initialization data for the transformer.	
Available via	ComXf.h	

⌋*(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] ⌈

Service Name	ComXf_<transformerId>	

Syntax	`uint8 ComXf_<transformerId> (`
	`  uint8* buffer,`
	`  uint32* bufferLength,`
	`  <paramtype> dataElement`
	`)`	

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

⌋*(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 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])
- `transformerId` is the name pattern for the transformer specified be 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] ⌈

Service Name	ComXf_Inv_<transformerId>	

Syntax	`uint8 ComXf_Inv_<transformerId> (`
	`  const uint8* buffer,`
	`  uint32 bufferLength,`
	`  <type>* dataElement`
	`)`	

Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	buffer	Buffer allocated by the RTE, where the still serialized data are stored by the Rte
	bufferLength	Used length of the buffer
Parameters (inout)	None	
Parameters (out)	dataElement	Data element which is the result of the transformation and contains the deserialized data element
Return value	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
Description	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.	
Available via	ComXf.h	

⌋*(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] ⌈

Service Name	ComXf_Init	

Syntax	`void ComXf_Init (`
	`  const ComXf_ConfigType* config`
	`)`	

Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	config	Pointer to the transformer's configuration data.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This service initializes the transformer for the further processing.	
Available via	ComXf.h	

⌋ *(SRS_BSW_00407, SRS_BSW_00411)*

8.3.4 ComXf_DeInit

[SWS_ComXf_00027] ⌈

Service Name	ComXf_DeInit

Syntax	`void ComXf_DeInit (`
	`  void`
	`)`

Service ID [hex]	0x02
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	This service deinitializes the transformer.
Available via	ComXf.h

⌋ *(SRS_BSW_00407, SRS_BSW_00411)*

8.3.5 ComXf_GetVersionInfo

[SWS_ComXf_00024] ⌈

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

⌋(*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_Xfrm_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 Xfrm configuration [2] - may be extended by the vendor to support the close interaction with the Com module [4].⌋ *(SRS_Xfrm_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 `ComSignalGroup`s.⌋ *(SRS_Xfrm_00202)*

[SWS_ComXf_00025] ⌈The `apiServicePrefix` of the COM Based Transformer's EcuC shall be set to `ComXf`.⌋ *(SRS_BSW_00159, SRS_Xfrm_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.

Class	**DataTransformation**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.dataTransformation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data Transformation Kind	DataTransformationKind Enum	0..1	attr	This attribute controls the kind of DataTransformation to be applied.
executeDespite Data Unavailability	Boolean	1	attr	Specifies whether the transformer chain is executed even if no input data are available.
transformer Chain (ordered)	Transformation Technology	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.1: 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. **Tags:**atp.recommendedPackage=EcucModuleDefs			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpDefinition*, *CollectableElement*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
apiServicePrefix	CIdentifier	0..1	attr	For modules where several instances of the VSMD can be defined the apiServicePrefix defines the API namespace of the derived instances, e.g. Cdd, Xfrm (ComXf, SomeIpXf, E2EXf).
container	EcucContainerDef	*	aggr	Aggregates the top-level container definitions of this specific module definition. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=container.shortName xml.sequenceOffset=11
postBuildVariant Support	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.

▽

△

Class	**EcucModuleDef**			
refinedModule Def	EcucModuleDef	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. **Stereotypes:** atpUriDef
supported ConfigVariant	EcucConfiguration VariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the Ecuc ModuleDef 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.

Table A.2: EcucModuleDef

Class	**ISignal**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup. **Tags:**atp.recommendedPackage=ISignals			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
data Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataTransformation.dataTransformation, dataTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	1	attr	With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.

▽

△

Class	ISignal			
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "Init Value". If a full DataMapping exist for the SystemSignal this information may be available from a configured Sender ComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification.
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=iSignalProps
iSignalType	ISignalTypeEnum	0..1	attr	This attribute defines whether this iSignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type.
length	UnlimitedInteger	1	attr	Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalid Value" and the Data Semantics. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=networkRepresentationProps
systemSignal	SystemSignal	1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal.
timeout Substitution Value	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstituition for this ISignal.

▽

△

Class	**ISignal**			
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=transformationISignalProps

Table A.3: ISignal

Class	**ISignalGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	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. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage) **Tags:**atp.recommendedPackage=ISignalGroup			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comBased SignalGroup Transformation	DataTransformation	0..1	ref	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation.data Transformation, comBasedSignalGroup Transformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together.
systemSignal Group	SystemSignalGroup	1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup.
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignal Groups are described in the TransformationTechnology class. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=transformationISignalProps

Table A.4: ISignalGroup

Class	**ISignalIPdu**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by Com. The ISignalIPdu 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. A maximum of one dynamic length signal per IPdu is allowed. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduTiming Specification	IPduTiming	0..1	aggr	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. atpVariation: The timing of a Pdu can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iPduTimingSpecification, iPduTiming Specification.variationPoint.shortLabel vh.latestBindingTime=postBuild
iSignalToPdu Mapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the Signal IPdu. atpVariation: The content of a PDU can be variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iSignalToPduMapping.shortName, iSignalTo PduMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
unusedBit Pattern	Integer	1	attr	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.

Table A.5: ISignalIPdu

Class	**ISignalToIPduMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ISignalIPdu.iSignalToPduMapping, NmPdu.iSignalToIPduMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iSignal	ISignal	0..1	ref	Reference to a ISignal that is mapped into the ISignal IPdu. 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.

▽

△

Class	**ISignalToIPduMapping**			
iSignalGroup	ISignalGroup	0..1	ref	Reference to an ISignalGroup that is mapped into the SignalIPdu. If an ISignalToIPduMapping for an ISignal Group is defined, only the UpdateIndicationBitPosition and the transferProperty is relevant. The startPosition and the packingByteOrder shall be ignored. 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.
packingByte Order	ByteOrderEnum	0..1	attr	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). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.
startPosition	UnlimitedInteger	0..1	attr	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. 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. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.
transferProperty	TransferPropertyEnum	0..1	attr	Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.
update IndicationBit Position	UnlimitedInteger	0..1	attr	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. Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. 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.

Table A.6: ISignalToIPduMapping

Class	**ImplementationDataType**			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code. **Tags:**atp.recommendedPackage=ImplementationDataTypes			
Base	*ARElement*, *ARObject*, *AbstractImplementationDataType*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
isStructWith Optional Element	Boolean	0..1	attr	This attribute is only valid if the attribute category is set to STRUCTURE. If set to true, this attribute indicates that the ImplementationDataType has been created with the intention to define at least one element of the structure as optional.
subElement (ordered)	ImplementationData TypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of ImplementionDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a Implementation DataType representing a structure. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=subElement.shortName, sub Element.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the Implementation DataType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=symbolProps.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.7: ImplementationDataType

Class	***PortPrototype*** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	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.			
Base	*ARObject*, *AtpBlueprintable*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/ server communication.
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port.

▽

△

Class	***PortPrototype*** (abstract)			
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port.
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port.
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table A.8: PortPrototype

Class	**SenderReceiverInterface**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A sender/receiver interface declares a number of data elements to be sent and received. **Tags:**atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	aggr	The data elements of this SenderReceiverInterface.
invalidation Policy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement
metaDataItem Set	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with dataElements of the enclosing Sender ReceiverInterface

Table A.9: SenderReceiverInterface

Class	**SenderReceiverToSignalGroupMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1	iref	Reference to a data element with a composite datatype which is mapped to a signal group. **InstanceRef implemented by:**VariableDataPrototypeIn SystemInstanceRef
signalGroup	SystemSignalGroup	1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type
typeMapping	SenderRecComposite TypeMapping	1	aggr	The CompositeTypeMapping maps the ApplicationArray Elements and ApplicationRecordElements to Signals of the SignalGroup.

Table A.10: SenderReceiverToSignalGroupMapping

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

Table A.11: SystemSignalGroup

B Used ECU Configuration

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

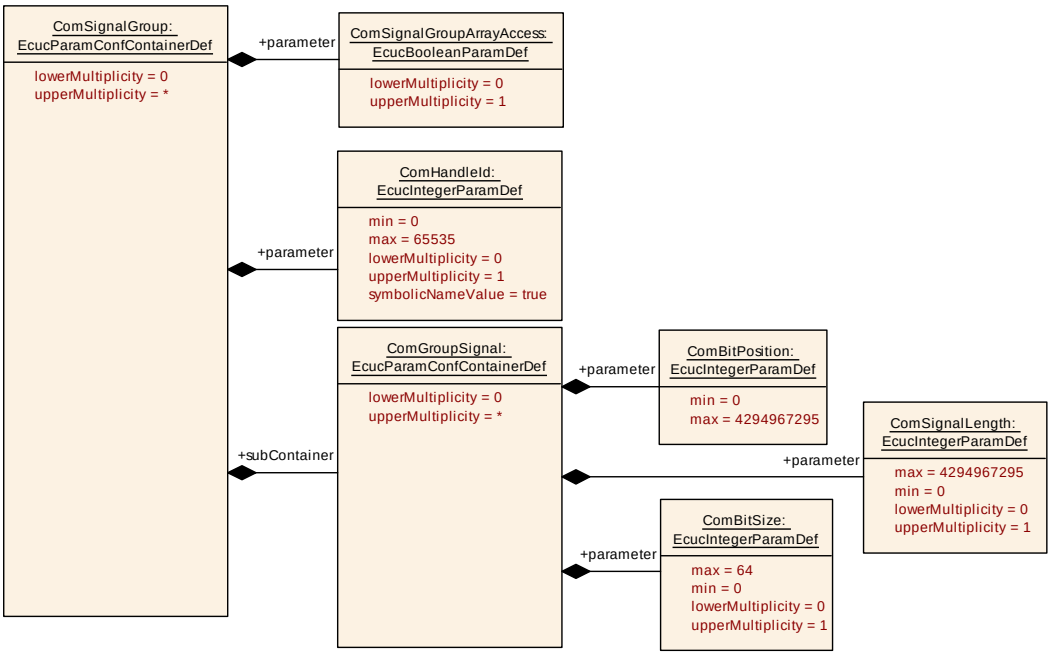

Figure B.1: AR_EcucDef_ComSignalGroupArrayAccess

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

SWS Item	**[ECUC_Com_00314]**	
Parameter Name	ComDataInvalidAction	
Parent Container	ComSignalGroup	
Description	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.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	NOTIFY	–
	REPLACE	Literal for DataInvalidAction
Post-Build Variant Multiplicity	false	

▽

△

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

SWS Item	**[ECUC_Com_00183]**		
Parameter Name	ComFirstTimeout		
Parent Container	ComSignalGroup		
Description	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.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00165]**		
Parameter Name	ComHandleId		
Parent Container	ComSignalGroup		
Description	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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	

▽

△

	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_Com_00811]**		
Parameter Name	ComInitialValueOnly		
Parent Container	ComSignalGroup		
Description	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 users (e.g. RTE, SwCluC). Thus the Com implementation does not need to expect any API calls for this signal (group).		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00412]**		
Parameter Name	ComRxDataTimeoutAction		
Parent Container	ComSignalGroup		
Description	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInit Value	
	SUBSTITUTE	signals shall be replaced by their ComTimeout SubstitutionValue	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

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

SWS Item	[ECUC_Com_00263]		
Parameter Name	ComTimeout		
Parent Container	ComSignalGroup		
Description	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.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	[ECUC_Com_00232]	
Parameter Name	ComTransferProperty	
Parent Container	ComSignalGroup	
Description	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.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	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.	
	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.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00257]**		
Parameter Name	ComUpdateBitPosition		
Parent Container	ComSignalGroup		
Description	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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

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

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

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

SWS Item	[ECUC_Com_00259]		
Parameter Name	ComBitPosition		
Parent Container	ComGroupSignal		
Description	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		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

△

Scope / Dependency	scope: local

SWS Item	**[ECUC_Com_00158]**		
Parameter Name	ComBitSize		
Parent Container	ComGroupSignal		
Description	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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 64		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00165]**		
Parameter Name	ComHandleId		
Parent Container	ComGroupSignal		
Description	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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_Com_00391]**
Parameter Name	ComSignalDataInvalidValue
Parent Container	ComGroupSignal

▽

△

Description	Defines the data invalid value of the signal. 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.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

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

SWS Item	**[ECUC_Com_00170]**
Parameter Name	ComSignalInitValue
Parent Container	ComGroupSignal

▽

△

Description	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. 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.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	0		
Regular Expression	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

SWS Item	**[ECUC_Com_00437]**		
Parameter Name	ComSignalLength		
Parent Container	ComGroupSignal		
Description	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. 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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE

▽

△

	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00127]**		
Parameter Name	ComSignalType		
Parent Container	ComGroupSignal		
Description	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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN	–	
	FLOAT32	–	
	FLOAT64	–	
	SINT16	–	
	SINT32	–	
	SINT64	–	
	SINT8	–	
	UINT16	–	
	UINT32	–	
	UINT64	–	
	UINT8	–	
	UINT8_DYN	–	
	UINT8_N	–	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_10006]**
Parameter Name	ComTimeoutSubstitutionValue
Parent Container	ComGroupSignal
Description	The signal substitution value will be used in case of a timeout and ComRxDataTimeout Action is set to SUBSTITUTE. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case ofUINT8_DYN the initial size shall be 0. 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.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00560]**		
Parameter Name	ComTransferProperty		
Parent Container	ComGroupSignal		
Description	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.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	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.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00002]**
Parameter Name	ComSystemTemplateSystemSignalRef
Parent Container	ComGroupSignal
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.

▽

△

Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	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.

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

SWS Item	**[ECUC_Com_00387]**		
Parameter Name	ComIPduCallout		
Parent Container	ComIPdu		
Description	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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

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

SWS Item	[ECUC_Com_00493]		
Parameter Name	ComIPduDirection		
Parent Container	ComIPdu		
Description	The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	RECEIVE	–	
	SEND	–	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC_Com_00496		

SWS Item	[ECUC_Com_00175]
Parameter Name	ComIPduHandleId
Parent Container	ComIPdu

▽

△

Description	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_Trigger Transmit, Com_TriggerIPDUSend or Com_TriggerIPDUSendWithMetaData, Com_Copy TxData 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 ComIPdu Callout and/or ComIPduTriggerTransmitCallout.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_Com_00119]**		
Parameter Name	ComIPduSignalProcessing		
Parent Container	ComIPdu		
Description	For the definition of the two modes Immediate and Deferred.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DEFERRED	signal indication / confirmations are deferred for example to a cyclic task	
	IMMEDIATE	the signal indications / confirmations are performed in Com_RxIndication/ Com_Tx Confirmation	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00765]**
Parameter Name	ComIPduTriggerTransmitCallout
Parent Container	ComIPdu
Description	If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Post-Build Variant Multiplicity	false

▽

△

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

SWS Item	**[ECUC_Com_00761]**		
Parameter Name	ComIPduType		
Parent Container	ComIPdu		
Description	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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	NORMAL	sent or received via normal L-PDU	
	TP	sent or received via TP	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

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

SWS Item	**[ECUC_Com_10012]**
Parameter Name	ComIPduMainFunctionRef
Parent Container	ComIPdu
Description	Reference to the Com_MainFunctionRx/Com_MainFunctionTx this I-PDU belongs to. Mandatory, if multiple main functions of the relevant type are defined.
Multiplicity	0..1
Type	Choice reference to [ComMainFunctionRx, ComMainFunctionTx]

▽

△

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

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

SWS Item	**[ECUC_Com_00518]**		
Parameter Name	ComIPduSignalRef		
Parent Container	ComIPdu		
Description	References to all signals contained in this I-PDU.		
Multiplicity	0..*		
Type	Reference to ComSignal		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_10021]**
Parameter Name	ComMainFunctionRouteSignalsRef
Parent Container	ComIPdu

▽

△

Description	Reference to ComMainFunctionRouteSignals which performs signal gateway related activities.		
Multiplicity	0..1		
Type	Reference to ComMainFunctionRouteSignals		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

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

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
ComTxIPdu	0..1	This container must be included if COM_IPDU_DIRECTION is configured to SEND.

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

SWS Item	**[ECUC_Com_00181]**
Parameter Name	ComMinimumDelayTime
Parent Container	ComTxIPdu

▽

△

Description	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_Trigger Transmit API.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 3600]		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00576]**		
Parameter Name	ComTxIPduClearUpdateBit		
Parent Container	ComTxIPdu		
Description	Defines when the update-bits of signals or signal groups, contained in this I-PDU, will be cleared.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	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.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**[ECUC_Com_00017]**
Parameter Name	ComTxIPduUnusedAreasDefault
Parent Container	ComTxIPdu

▽

△

Description	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.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
ComMetaDataDefaultItem	0..*	Defines a default value for a meta data item. Used for sending an I-PDU with meta data when it is triggered spontaneously (and not by Com_TriggerIPDUSendWithMetaData), and no meta data has been provided by the RTE. It represents a MetaDataItem of the referenced global PDU.
ComTxModeFalse	0..1	The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to false. The default is transmission mode None.
ComTxModeTrue	0..1	The referenced transmission mode object that is used when the filtering state for this I-PDU 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

C.5 Constraint History of this Document according to AUTOSAR R4.4.0

C.5.1 Added Traceables in 4.4.0

none

C.5.2 Changed Traceables in 4.4.0

[SWS_ComXf_00007] [SWS_ComXf_00020]

C.5.3 Deleted Traceables in 4.4.0

[SWS_ComXf_00001] [SWS_ComXf_00014]

C.6 Constraint History of this Document according to AUTOSAR R19-11

C.6.1 Added Traceables in 19-11

none

C.6.2 Changed Traceables in 19-11

none

C.6.3 Deleted Traceables in 19-11

none

C.7 Constraint History of this Document according to AUTOSAR R20-11

C.7.1 Added Traceables in R20-11

none

C.7.2 Changed Traceables in R20-11

none

C.7.3 Deleted Traceables in R20-11

none

C.8 Constraint History of this Document according to AUTOSAR R21-11

C.8.1 Added Traceables in R21-11

[SWS_ComXf_00037]

C.8.2 Changed Traceables in R21-11

none

C.8.3 Deleted Traceables in R21-11

none

C.9 Constraint History of this Document according to AUTOSAR R22-11

C.9.1 Added Traceables in R22-11

none

C.9.2 Changed Traceables in R22-11

none

C.9.3 Deleted Traceables in R22-11

none