

Document Title	System Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	63

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R19-11

Document Change History			
Date	Release	Changed by	Description
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Rework of Ethernet communication model • Added support for Signal-To-Service Translation • Added support for IPsec configuration • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Added support for BusMirroring • Reworked the modeling of LinSlaves • Introduced Crypto Infrastructure for SecuredIPdu • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Added support for new E2E Profiles 7, 11 and 22 • Improved configuration of Ethernet Switch Ports • Introduced Security Profiles • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • Introduction of data transformation • Introduction of SecuredIPdu • Introduction of Switch Configuration • Introduction of Global Time Synchronization • Improved support for CanFD • Minor corrections / clarifications / editorial changes; For details please refer to the BWCStatement
2014-03-31	4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> • Various fixes and clarifications
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> • Set CanNmCluster.nmChannelActive, FlexrayArTpChannel.timeFrlf and FlexrayArTpChannel.maxFrlf to deprecated • Added SoAd Pdu Collection attributes to SocketConnection • Added SoAdRouting-Group.eventGroupControlType • Introduced SocketAddress.multicastConnector • Clarified usage of ISignal.dataTypePolicy • Described the handling of ComSpecs during flattening • Introduced new Pdu types: GeneralPurposePdu and GeneralPurposeIPdu
			<ul style="list-style-type: none"> • Made RootSwCompositionPrototype.calibrationParameterValueSet "atpSplittable" • Made RootSwCompositionPrototype.flatMap "atpSplittable" • Added new Ethernet addressing attributes to SocketConnection to help to derive the Ecu Configurations for the Server and the Clients

2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Added support for remote activation of RunnableEntitys • Added support VLANs and Service Discovery • Reworked the SoAd configuration • Introduced SenderReceiverCompositeElementToSignalMapping and ClientServerToSignalMapping • Added support for CAN FD • Reworked the J1939 TP configuration • Clarification of the usage of swDataDefProps on ISignals and SystemSignals • Added support for Complex Drivers in the Topology • Updated IPduM to allow only static part reception • Added LinSlaveConfig class to the LinMaster • Clarified meaning of PduToFrameMapping.startPosition
2011-12-22	4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> • Added support for Partial Networking • Added support for Complex Drivers • Added support for new COM transfer properties • Added support for transmission mode switch via Com_SwitchIpduTxMode COM API • Added support for treating byte arrays with primitive type mapping • Added support for partial routing in signal gateways • Added support for FlexRay AUTOSAR TP • Added rules for creation of Pdu Triggerings and Pdu Ports • Explained the general approach of bit counting

2009-12-18	4.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> • updated System class category names • Changed specification of PduLength parameter from bits to bytes • Made Flexray channel specific attributes optional • Clarified the usage of EcuPorts in System Extract/Ecu Extract • Allowed to define sending and receiving connections to EcuPorts for NmPdus, XcpPdus • Aligned FrTP model to AUTOSAR FrTp SWS • Replaced ComProcessingPeriod by three timebase parameters • Reworked E2E protection of selected I-PDUs • Corrected AssignFrameIdRange configuration in LIN model • Clarified the routing of ISignalGroups in the Signal Gateway • Extended the enumeration "TransferPropertyEnum" with the element "triggeredOnChange" • Added a subchapter to the appendix about special use cases that are supported by the System Template • Reworked SenderReceiverToSignalGroupMapping and ClientServerToSignalGroupMapping • Changed multiplicity between System and SystemMapping from 1 to 0..1.
------------	-------	------------------------	--

2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> ● Implemented support for LIN 2.1 ● Implemented support for Network Management (FlexRayNm, CanNm, LinNm, UdpNm) ● adapted IPdu Multiplexer model to ASAM Fibex 3.1 ● Reworked "ECU Extract" chapter ● Introduced "System Extract" ● Introduced EndToEndProtection for ISignalPdu's ● Reworked "Transport Layer" chapter ● Implemented Variant Handling concept ● Implemented Documentation support concept ● Implemented support for J1939 communication ● Implemented support for TTCan ● Implemented support for for TCP/IP and DoIP. ● Introduced Pdu Counter and Pdu Replication ● Implemented VMM/AMM concept ● Introduced low-level routing of NPdu's ● Implemented support for dynamic signals ● Introduced PdurIPduGroups
2009-02-04	3.1.2	AUTOSAR Administration	<ul style="list-style-type: none"> ● Clarified semantics of Data Mappings ● Added inheritance from Identifiable to PduToFrameMapping ● Added "FlexRayChannelName" attribute to FlexRayPhysicalChannel element.

2008-08-13	3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Added the boolean attribute "payloadPreambleIndicator" to the "FlexrayFrameTriggering". • Added extension that allows the assignment of IPduGroups to ECUs. • Added missing reference from "ClientServerComposite-TypeMapping" to "ArgumentPrototype" • Alignment with AUTOSAR IPduM SWS
2008-02-01	3.0.2	AUTOSAR Administration	<ul style="list-style-type: none"> • Moved "canAddressingMode" attribute from "CanCluster" to the "CanFrameTriggering" element • Clarified the descriptions of several elements and attributes.
2007-12-21	3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Communication part reworked from scratch • Alignment with ECU Configuration • Added support for Transport Protocols • Major changes in Topology chapter after harmonisation with Fibex (removed complex Topologies) • Document meta information extended • Small layout adaptations made

2006-11-28	2.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Support for Signal Groups added. • Rework of the Topology Description • Introduction of PDUs. Description of the PDU Multiplexer, PDU Gateway. • FlexRay: multiple transmission of a frame within one communication cycle is supported now. • Removed the concept of Variant Descriptions (Properties) and CompToECUMappingConstraints relying on the property concept. • Split SwCompToEcuMapping in two classes in order to allow separation of SWC-to-ECU mapping and Implementation-to-SWC mapping. • Removed preliminary chapter on MOST as it is not part of the standard. • For all Instance References in the System Template added diagrams to the meta-model containing detailed representations of these references.
2005-05-31	1.0	AUTOSAR Administration	Initial Release

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	24
1.1	Abbreviations	24
1.2	Requirements Tracing	25
1.3	Requirements not fulfilled by TPS requirements	27
1.4	Methodology for Defining Formal Template	29
1.5	Scope	30
1.6	UML Meta-Model	33
1.7	Document Conventions	35
1.7.1	Detailed Representation of InstanceRef Associations	38
1.7.2	Variant Handling	38
1.7.3	Timing Extensions	38
1.7.4	Documentation Support	38
1.7.5	Stereotype atpSplittable in the System Template	40
1.8	AUTOSAR System Template and ASAM FIBEX	40
2	System	42
2.1	ClientIdDefinitionSet	45
3	Topology	47
3.1	ECUs and their communication capabilities	48
3.1.1	ECU Instance	49
3.1.2	Communication Controller	51
3.1.3	Communication Connector	52
3.2	Communication Clustering	54
3.2.1	Communication Cluster	54
3.2.2	Physical Channel	55
3.3	Specialized Attributes of the Topology Entities	56
3.3.1	CAN	58
3.3.1.1	CAN Cluster	59
3.3.1.2	CAN Communication Controller	60
3.3.1.3	CAN Physical Channel	65
3.3.1.4	CAN Communication Connector	65
3.3.2	TTCAN	66
3.3.2.1	TTCAN Cluster	67
3.3.2.2	TTCAN Communication Controller	68
3.3.2.3	TTCAN Physical Channel	69
3.3.2.4	TTCAN Communication Connector	69
3.3.3	SAE J1939	70
3.3.4	FlexRay	71
3.3.4.1	FlexRay Cluster	71
3.3.4.2	FlexRay Communication Controller	74
3.3.4.3	FlexRay Communication Connector	78
3.3.4.4	FlexRay Physical Channel	78
3.3.5	LIN	80

3.3.5.1	LIN Cluster	81
3.3.5.2	LIN Communication Controller	81
3.3.5.3	LIN Master	82
3.3.5.4	LIN Slave	84
3.3.5.5	LIN Communication Connector	86
3.3.5.6	LIN Physical Channel	87
3.3.6	Ethernet	88
3.3.6.1	Ethernet Cluster	90
3.3.6.2	Ethernet Physical Channel	91
3.3.6.3	Ethernet Coupling Elements and Coupling Ports	93
3.3.6.4	Ethernet Communication Controller	100
3.3.6.5	Ethernet Communication Connector	101
3.3.6.6	Ethernet Switch Driver	102
3.3.7	CDD	116
3.4	Mapping of Topology Entities onto Hardware Elements	117
3.4.1	ECU Mapping	118
3.4.2	Communication Controller Mapping	119
3.4.3	HW-Port Mapping	120
4	Top-level Software Composition	121
5	Mapping	124
5.1	Software Component Mapping	127
5.1.1	SW Component to ECU Mapping	127
5.1.2	Software Component to Implementation Mapping	131
5.1.3	SW Component to Partition Mapping	132
5.1.4	Software Component Mapping Constraints	135
5.1.4.1	ComponentClustering	136
5.1.4.2	ComponentSeparation	137
5.1.5	J1939 Controller Application Mapping	139
5.2	Data Mapping	140
5.2.1	Mapping of Variable Data Prototypes on System Signals	144
5.2.1.1	Mapping of Variable Data Prototypes with primitive datatypes on System Signals (Sender-Receiver Communication)	147
5.2.1.2	Mapping of Variable Data Prototypes with composite datatypes (Sender-Receiver Communication)	151
5.2.1.3	Mapping of Client Server Operations to System Signals	158
5.2.1.4	Mapping of a ApplicationCompositeElementDataPrototype within a composite application data type on a System Signal (Sender-Receiver Communication)	160
5.2.1.5	Mapping of Trigger to SystemSignal	161
5.2.2	Signal Path Constraint	163
5.2.2.1	CommonSignalPath	165
5.2.2.2	ForbiddenSignalPath	168
5.2.2.3	PermissibleSignalPath	169
5.2.2.4	SeparateSignalPath	170

5.3	RTE and basic software resource estimations	171
5.4	Partial Networking	174
5.4.1	Partial Networking and managed Ethernet switch	179
5.4.2	Partial Network Gateway	182
5.5	Com Management Mapping	182
6	Communication	184
6.1	Triggerings and Ports	186
6.2	ISignals	198
6.2.1	Efficient COM for large data	217
6.2.2	Big Endian and Little Endian memory layout of PduS and Frames	221
6.3	PDUUs	225
6.3.1	ContainerIPdu	238
6.3.2	SecuredIPdu	248
6.3.2.1	Crypto Infrastructure for SecuredIPdu	257
6.3.3	EndToEndProtection for ISignalGroups	264
6.3.4	GeneralPurposeConnection	270
6.4	IPdu Timing	271
6.4.1	Data Filter configuration	276
6.4.2	Cyclic Timing	278
6.4.3	EventControlled Timing	279
6.4.4	Configuration of a trigger for COM_TriggerIPduSend API call	281
6.5	I-Pdu Multiplexer	283
6.5.1	I-Pdu Multiplexer in System Extract/ECU Extract	292
6.6	Frames	294
6.7	Specialized Attributes of the Communication Entities	296
6.7.1	FlexRay specific description	296
6.7.2	LIN specific description	302
6.7.2.1	LIN Frames	303
6.7.2.2	LIN Schedule Table	307
6.7.2.3	Configuration Services	310
6.7.3	CAN specific description	315
6.7.3.1	SAE J1939 Protocol specific description	319
6.7.4	TTCAN specific description	320
6.7.5	Ethernet specific description	322
6.7.5.1	ApplicationEndpoint	327
6.7.5.2	NetworkEndpoint	332
6.7.5.3	SOME/IP Service Instances	340
6.7.5.4	ProvidedServiceInstance	343
6.7.5.5	ConsumedServiceInstance	357
6.7.5.6	Service Discovery Server Configuration	367
6.7.5.7	Service Discovery Client Configuration	372
6.7.5.8	Group of ConsumedServiceInstances and ProvidedServiceInstances	375
6.7.5.9	StaticSocketConnection	377

6.7.5.10	Service Discovery Message Configuration	381
6.7.5.11	NmPdu data exchange over StaticSocketConnection	383
6.7.5.12	Diagnostics over IP	385
6.7.5.13	Transport Layer Security	389
6.7.5.14	IPsec	396
6.7.5.15	EthernetFrameType based communication	404
6.8	Transport Layer	407
6.8.1	Transport Layer Routing	409
6.8.2	FlexRay ISO Transport Layer	409
6.8.3	FlexRay AUTOSAR Transport Layer	416
6.8.4	CAN Transport Layer	422
6.8.5	LIN Transport Layer	429
6.8.6	Ethernet Transport Layer	434
6.8.7	SOME/IP segmenter	435
6.8.8	SAE J1939 Transport Layer	438
6.8.9	Unicast TP Example	443
6.8.10	Multicast TP Example	444
6.8.11	Diagnostic Connection	446
6.9	Network Management	450
6.9.1	FlexRay Network Management	458
6.9.2	CAN Network Management	461
6.9.3	UDP Network Management	465
6.9.4	J1939 Network Management	469
6.9.4.1	J1939SharedAddressCluster	471
6.9.5	Managed Channels	473
6.10	Bus Mirroring	473
6.10.1	CAN Destination Channel	476
6.10.2	FlexRay Destination Channel	478
6.10.3	Ethernet Destination Channel	480
6.10.4	User Defined Destination Channel	481
6.11	Fan-out	482
6.11.1	Signal fan-out	482
6.11.1.1	RTE fan-out	482
6.11.1.2	COM Signal Gateway fan-out	482
6.11.2	Pdu fan-out	483
6.11.2.1	Pdu Router fan-out	483
6.11.2.2	Flexray Interface fan-out	484
6.11.3	Frame fan-out	485
6.12	Log and Trace	485
6.13	Support of Complex Drivers	491
6.14	MetaData in SenderReceiverInterface	492
6.15	Signal Service Translation	493
6.15.1	Architectural setup	494
6.15.1.1	Method handling	495
6.15.2	Mapping description	496
6.15.2.1	Mapping description from several sources	500

6.15.2.2	Mapping description and data conversion	503
6.15.3	Service discovery control	503
6.15.3.1	Service control at ECU start	504
6.15.3.2	Service control due to availability of partial networks	505
6.15.3.3	Service control due to availability of related service instance	506
6.15.4	Translation behavior	507
6.15.4.1	COM-Stack translation behavior	507
6.15.4.2	Translation Application Software Component translation behavior	507
6.15.5	End-to-End considerations	508
6.15.5.1	Safety	508
6.15.5.2	Security	512
7	Data Transformation	513
7.1	Outline	513
7.1.1	Configuration of the Communication Layout	513
7.1.2	Data Transformation by Software	514
7.2	Use Cases	515
7.2.1	Transmission of large composite data types over networks with large PDUs (e.g Ethernet)	515
7.2.2	Support of transmission from one sender to multiple receivers with Signal Fan-out	516
7.2.3	Support of transmission from one sender to multiple receivers with PDU Fan-out	516
7.2.4	Transformer Chaining	517
7.2.5	Signal Group Based interaction of the transformer with the Com module	518
7.3	Transformer configuration	518
7.3.1	Generic Transformer	533
7.3.2	SOME/IP Transformer	534
7.3.2.1	Handling of Strings	539
7.3.2.2	SOME/IP Transformation Properties on the level of DataPrototypes	540
7.3.2.3	Network Representation	548
7.3.3	COM Based Transformer	557
7.3.4	E2E Transformer	561
7.3.4.1	E2E state machine settings	573
7.3.4.2	E2E recommended configuration settings	573
7.3.5	UserDefined Transformer	580
7.3.6	Support for TLV Encoding	581
7.3.6.1	Assignment of TLV Data Ids	581
7.3.6.2	Assignment of TLV Wire Type	586
8	Gateways	587
8.1	Frame Mapping	589
8.2	IPdu Mapping	590

8.2.1	Routing and processing of Diagnostics Pdus	593
8.3	Signal Mapping	594
8.3.1	Partial Signal Group Mapping	595
9	Global Time Synchronization	597
9.1	Introduction	597
9.2	The big Picture	598
9.3	Detailed Description of Global Time Synchronization	608
9.3.1	Time Synchronization over CAN	609
9.3.2	Time Synchronization over Ethernet	611
9.3.2.1	Time Synchronization and Ethernet propagation delay	615
9.3.2.2	Time Synchronization and Ethernet connection	615
9.3.2.3	Time Synchronization and managed Ethernet switch	616
9.3.3	Time Synchronization over FlexRay	620
9.3.4	Time Synchronization by user defined Timebase Provider	622
9.3.5	Time Synchronization Common Properties	623
10	Usage of the System Template	625
10.1	System Constraint Description	626
10.2	Abstract System Description	628
11	System Extract of the System Configuration Description	630
11.1	OEM/Supplier Collaboration Scenario	631
11.2	Data Mapping in the System Extract	633
11.3	SW component inclusion and top level data mapping	636
11.4	Port-Interface Mapping in the System Extract	638
12	ECU Extract of the System Configuration Description	639
12.1	Topology	641
12.2	Top-level Software Composition	641
12.2.1	ECU Flat view	644
12.2.2	Internal Communication	644
12.2.3	External Communication	646
12.2.4	Port Groups	651
12.2.5	Service Needs	652
12.3	Communication	653
12.3.1	Frame	653
12.3.2	PDU	654
12.3.3	ISignals and ISignalGroups	654
12.3.4	SystemSignal and SystemSignalGroup	655
12.3.5	Gateways	655
12.3.6	TP configuration	656
12.3.7	NM configuration	656
12.4	Naming Issues	656
12.4.1	Package Structure	656
12.4.2	Naming of Measurement and Calibration Data	657
12.4.3	Naming of Derived Elements	658

12.4.4	Re-use of short names assigned in previous iterations	659
12.5	ECU Extract in subsequent Cycles of Iterative Development	659
12.5.1	Traceability of model elements created in ECU Extract	659
12.5.2	Mapping of AUTOSAR attributes to ASAM ASAP2	664
12.5.3	Assigning communication graphs to RTE Implementation Plug-Ins	665
12.6	Variant Handling in ECU Extract	667
12.6.1	System Constants	667
12.6.2	Nested Whole/Part class variants	668
12.6.3	Multiple instances of calibration parameters in system scope	669
13	Supported special use-cases	670
13.1	Support of sending / receiving same Can/Flexray Frame on same chan- nel (Pdu Gateway Use-Case)	670
13.2	Support of sending / receiving same Can/Flexray Frame on same chan- nel (bidirectional routing in COM)	671
13.3	Support of dynamic CAN IDs	674
13.4	N:1 Sender Receiver communication description in a System Extract over one <code>PhysicalChannel</code>	674
13.5	Description of MOST Functions	677
A	Glossary	679
B	Detailed Representation of InstanceRef Associations in the System Template	683
B.1	Usage of InstanceRefs in Data Mapping diagrams	683
B.2	Usage of InstanceRefs in SW Mapping diagrams	684
B.3	Usage of InstanceRefs in Signal Path Constraint diagrams	685
B.4	Usage of InstanceRefs in PncMapping	685
B.5	Usage of InstanceRefs in ComManagementMapping	686
B.6	"SWC in System" InstanceRef	687
B.7	"Operation in System" InstanceRef	689
B.8	"VariableDataPrototype" InstanceRef	691
B.9	"PortGroup in System" InstanceRef	695
B.10	"DataPrototype in PortInterface" InstanceRef	697
C	Harmonisation between Upstream Templates and ECU Configuration	699
C.1	ComStack	700
C.1.1	Com Mapping	700
C.1.2	LdCom Mapping	769
C.1.3	IPduM Mapping	774
C.1.4	SecOc Mapping	818
C.1.5	PduR	851
C.1.6	Nm Interface	870
C.1.7	EcuC	882
C.1.8	ComM	899
C.1.9	Xcp	917
C.1.10	Bus Mirroring	940

C.2	Can	979
C.2.1	Can Driver Mapping	979
C.2.2	Can Interface Mapping	1016
C.2.3	Can Transceiver Mapping	1062
C.2.4	CanNm Mapping	1079
C.2.5	CanTp Mapping	1102
C.2.6	CanSm Mapping	1123
C.3	J1939	1132
C.3.1	J1939Tp Mapping	1132
C.3.2	J1939Nm Mapping	1152
C.3.3	J1939Dcm Mapping	1160
C.3.4	J1939Rm Mapping	1162
C.4	FlexRay	1167
C.4.1	FlexRay Driver Mapping	1167
C.4.2	FlexRay Interface Mapping	1190
C.4.3	FrNm Mapping	1234
C.4.4	FrTp Mapping	1257
C.4.5	FrArTp Mapping	1273
C.4.6	FrSM Mapping	1289
C.5	Lin	1298
C.5.1	Lin Driver Mapping	1298
C.5.2	Lin Interface Mapping	1304
C.5.3	LinTp Mapping	1333
C.6	Ethernet	1342
C.6.1	Ethernet Driver Mapping	1342
C.6.2	Ethernet Interface Mapping	1361
C.6.3	Ethernet Switch Driver Mapping	1384
C.6.4	Service Discovery	1428
C.6.5	SoAd	1469
C.6.6	EthSM	1505
C.6.7	EthTrcv	1508
C.6.8	Tcplp	1526
C.6.9	DoIP	1632
C.6.10	UdpNm	1662
C.6.11	SomelpTp	1684
C.6.12	Wireless Ethernet Driver Mapping	1691
C.7	Diagnostic	1697
C.7.1	Dcm Mapping	1697
C.8	Time management	1698
C.8.1	StbM Time Management	1698
C.8.2	CAN Time Management	1713
C.8.3	Ethernet Time Management	1731
C.8.4	Flexray Time Management	1754
C.9	Crypto Stack	1769
C.9.1	CryptoDriver	1769
C.9.2	Crypto Service Manager	1795

C.10	Services	1800
C.10.1	Transformer General	1800
C.11	Miscellaneous	1802
C.11.1	Log and Trace	1802
C.11.2	ECU State Manager	1803
D	Constraint History	1805
D.1	Constraint History of this Document according to AUTOSAR R4.0.1	1805
D.1.1	Changed Constraints in R4.0.1	1805
D.1.2	Added Constraints in R4.0.1	1805
D.1.3	Deleted Constraints in R4.0.1	1805
D.2	Constraint History of this Document according to AUTOSAR R4.0.2	1805
D.2.1	Changed Constraints in R4.0.2	1805
D.2.2	Added Constraints in R4.0.2	1806
D.2.3	Deleted Constraints in R4.0.2	1806
D.3	Constraint and Specification Item History of this document according to AUTOSAR R4.0.3	1806
D.3.1	Changed Constraints in R4.0.3	1806
D.3.2	Changed Specification Items in R4.0.3	1806
D.3.3	Added Constraints in R4.0.3	1806
D.3.4	Added Specification Items in R4.0.3	1806
D.3.5	Deleted Constraints in R4.0.3	1807
D.3.6	Deleted Specification Items in R4.0.3	1807
D.4	Constraint and Specification Item History of this document according to AUTOSAR R4.1.1	1807
D.4.1	Changed Constraints in R4.1.1	1807
D.4.2	Changed Specification Items in R4.1.1	1807
D.4.3	Added Constraints in R4.1.1	1807
D.4.4	Added Specification Items in R4.1.1	1809
D.4.5	Deleted Constraints in R4.1.1	1812
D.4.6	Deleted Specification Items in R4.1.1	1812
D.5	Constraint and Specification Item History of this document according to AUTOSAR R4.1.2	1812
D.5.1	Changed Specification Items in R4.1.2	1812
D.5.2	Added Specification Items in R4.1.2	1812
D.5.3	Added Constraints in R4.1.2	1813
D.5.4	Changed Constraints in R4.1.2	1813
D.5.5	Deleted Constraints in R4.1.2	1813
D.6	Constraint and Specification Item History of this document according to AUTOSAR R4.1.3	1814
D.6.1	Changed Specification Items in R4.1.3	1814
D.6.2	Added Specification Items in R4.1.3	1814
D.6.3	Deleted Specification Items in R4.1.3	1814
D.6.4	Added Constraints in R4.1.3	1814
D.6.5	Changed Constraints in R4.1.3	1815
D.6.6	Deleted Constraints in R4.1.3	1815

D.7	Constraint and Specification Item History of this document according to AUTOSAR R4.2.1	1815
D.7.1	Added Traceables in 4.2.1	1815
D.7.2	Changed Traceables in 4.2.1	1817
D.7.3	Deleted Traceables in 4.2.1	1817
D.7.4	Added Constraints in 4.2.1	1818
D.7.5	Changed Constraints in 4.2.1	1821
D.7.6	Deleted Constraints in 4.2.1	1821
D.8	Constraint and Specification Item History of this document according to AUTOSAR R4.2.2	1821
D.8.1	Added Traceables in 4.2.2	1821
D.8.2	Changed Traceables in 4.2.2	1822
D.8.3	Deleted Traceables in 4.2.2	1823
D.8.4	Added Constraints in 4.2.2	1823
D.8.5	Changed Constraints in 4.2.2	1824
D.8.6	Deleted Constraints in 4.2.2	1825
D.9	Constraint and Specification Item History of this document according to AUTOSAR R4.3.0	1825
D.9.1	Added Traceables in 4.3.0	1825
D.9.2	Changed Traceables in 4.3.0	1827
D.9.3	Deleted Traceables in 4.3.0	1827
D.9.4	Added Constraints in 4.3.0	1828
D.9.5	Changed Constraints in 4.3.0	1830
D.9.6	Deleted Constraints in 4.3.0	1832
D.10	Constraint and Specification Item History of this document according to AUTOSAR R4.3.1	1832
D.10.1	Added Traceables in 4.3.1	1832
D.10.2	Changed Traceables in 4.3.1	1833
D.10.3	Deleted Traceables in 4.3.1	1833
D.10.4	Added Constraints in 4.3.1	1834
D.10.5	Changed Constraints in 4.3.1	1834
D.10.6	Deleted Constraints in 4.3.1	1835
D.11	Constraint and Specification Item History of this document according to AUTOSAR R4.4.0	1835
D.11.1	Added Traceables in 4.4.0	1835
D.11.2	Changed Traceables in 4.4.0	1836
D.11.3	Deleted Traceables in 4.4.0	1837
D.11.4	Added Constraints in 4.4.0	1837
D.11.5	Changed Constraints in 4.4.0	1839
D.11.6	Deleted Constraints in 4.4.0	1839
D.12	Constraint and Specification Item History of this document according to AUTOSAR R19-11	1840
D.12.1	Added Traceables in 19-11	1840
D.12.2	Changed Traceables in 19-11	1843
D.12.3	Deleted Traceables in 19-11	1844
D.12.4	Added Constraints in 19-11	1845

- D.12.5 Changed Constraints in 19-11 1847
- D.12.6 Deleted Constraints in 19-11 1849
- E Mentioned Class Tables 1850
- F Splitable Elements in the Scope of this Document 1925
- G Variation Points in the Scope of this Document 1926

References

- [1] Requirements on System Template
AUTOSAR_RS_SystemTemplate
- [2] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate
- [3] XML Schema Production Rules
AUTOSAR_TPS_XMLSchemaProductionRules
- [4] Methodology
AUTOSAR_TR_Methodology
- [5] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate
- [6] Specification of ECU Resource Template
AUTOSAR_TPS_ECUResourceTemplate
- [7] Standardization Template
AUTOSAR_TPS_StandardizationTemplate
- [8] Specification of Timing Extensions
AUTOSAR_TPS_TimingExtensions
- [9] ASAM Fibex – Field Bus Exchange Format, Version 3.1
<http://www.asam.net>
- [10] ISO 17987:2016 (all parts), Road vehicles – Local Interconnect Network (LIN)
<http://www.iso.org>
- [11] CAN specifications
<http://www.can-cia.org>
- [12] MOST Specification, Version 2.5
<http://www.mostnet.de>
- [13] FlexRay Protocol Specification
<http://www.flexray.com>
- [14] Serial Data Communications between Microcomputer Systems in heavy-duty Vehicle Applications
- [15] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture
- [16] Specification of LIN Interface
AUTOSAR_SWS_LINInterface
- [17] Specification of COM Based Transformer
AUTOSAR_SWS_COMBasedTransformer

- [18] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate
- [19] ASAM MCD 2MC ASAP2 Interface Specification
<http://www.asam.net>
ASAP2-V1.51.pdf
- [20] Specification of SW-C End-to-End Communication Protection Library
AUTOSAR_SWS_E2ELibrary
- [21] Specification of Communication
AUTOSAR_SWS_COM
- [22] Specification of I-PDU Multiplexer
AUTOSAR_SWS_IPDUMultiplexer
- [23] SAE J1939-21 Data Link Layer
- [24] Specification of Diagnostic over IP
AUTOSAR_SWS_DiagnosticOverIP
- [25] Specification of Cryptography for Adaptive Platform
AUTOSAR_SWS_Cryptography
- [26] Road vehicles – Diagnostics on Controller Area Networks (CAN) – Part2: Network layer services
- [27] Specification of Manifest
AUTOSAR_TPS_ManifestSpecification
- [28] Specification of SOME/IP Transformer
AUTOSAR_SWS_SOMEIPTransformer
- [29] Specification of Service Discovery
AUTOSAR_SWS_ServiceDiscovery
- [30] Specification of Secure Onboard Communication
AUTOSAR_SWS_SecureOnboardCommunication
- [31] Specification of TCP/IP Stack
AUTOSAR_SWS_Tcplp
- [32] Specification of RTE Software
AUTOSAR_SWS_RTE
- [33] SOME/IP Protocol Specification
AUTOSAR_PRS_SOMEIPProtocol
- [34] Specification of Module E2E Transformer
AUTOSAR_SWS_E2ETransformer
- [35] Specification of CRC Routines
AUTOSAR_SWS_CRCLibrary

- [36] E2E Protocol Specification
AUTOSAR_PRS_E2EProtocol
- [37] Specification of Synchronized Time-Base Manager
AUTOSAR_SWS_SynchronizedTimeBaseManager
- [38] Specification of Time Synchronization over Ethernet
AUTOSAR_SWS_TimeSyncOverEthernet
- [39] Software Process Engineering Meta-Model Specification
<http://www.omg.org/spec/SPEM/2.0/>

1 Introduction

1.1 Abbreviations

CAN	Controller Area Network
CAS	Collision Avoidance Symbol
CBV	Control Bit Vector
CC	Communication Controller
CMAC	Cipher-based message authentication code
DLC	Data Length Code
Dolp	Diagnostics over IP
DTD	Document Type Definition
ECU	Electrical Control Unit
FIBEX	Field Bus Exchange Format
I ² C	Inter-Integrated Circuit
ID	Identifier
IPDU	Interaction Layer Protocol Data Unit
ISG	Inter-slot Gap
LIN	Local Interconnect Network
LPDU	Data Link Layer Protocol Data Unit
MAC	Message Authentication Code
MAC Address	media access control address
MOST	Media Oriented Systems Transport
NAD	Node Address for Diagnostic
NID	NOde Identification
NIT	Network Idle Time
NM	Network Management
NPDU	Network Layer Protocol Data Unit
OBD	Onboard Diagnostic
PDU	Protocol Data Unit
POC	Protocol Operation Control
PSK	Pre-shared Key
RSA	Rivest-Shamir-Adleman. A method using public and private key for data encryption and decryption.
RTE	Runtime Environment
SDU	Service Data Unit
SID	Service Identifier
SPI	Serial Peripheral Interface
SWC	Software Component
SWC-T	Software Component Template
SYS-T	System Template
TLS	Transport Layer Security
TP	Transport Protocol
TTCAN	Time Triggered Controller Area Network
UML	Unified Modeling Language
VFB	Virtual Functional Bus
XML	Extensible Markup Language
XSD	XML Schema Definition

Table 1.1: Abbreviations used in the scope of this Document

1.2 Requirements Tracing

The following table references the requirements specified in [1] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_SYST_00001]	Mixed Systems (AUTOSAR/ NON-AUTOSAR)	[TPS_SYST_01063] [TPS_SYST_05000]
[RS_SYST_00002]	Basic Software Resources and RTE Resources	[TPS_SYST_01126]
[RS_SYST_00003]	Iterative Development	[TPS_SYST_01000] [TPS_SYST_01002] [TPS_SYST_01003]
[RS_SYST_00006]	Compatibility between the AUTOSAR Templates	[TPS_SYST_01017] [TPS_SYST_01019]
[RS_SYST_00007]	Mapping of Software Components to ECUs	[TPS_SYST_01001] [TPS_SYST_01020] [TPS_SYST_01021] [TPS_SYST_01022] [TPS_SYST_02114]
[RS_SYST_00008]	SWC Cluster	[TPS_SYST_01024] [TPS_SYST_01025]
[RS_SYST_00009]	SWC Separation	[TPS_SYST_01026] [TPS_SYST_01045]
[RS_SYST_00013]	Topology	[TPS_SYST_01004] [TPS_SYST_01005] [TPS_SYST_01006] [TPS_SYST_01007] [TPS_SYST_01008] [TPS_SYST_01009] [TPS_SYST_01010] [TPS_SYST_01011] [TPS_SYST_01013] [TPS_SYST_01014] [TPS_SYST_01015]
[RS_SYST_00014]	Data Segmentation	[TPS_SYST_01099] [TPS_SYST_01100] [TPS_SYST_01101] [TPS_SYST_01102] [TPS_SYST_01103] [TPS_SYST_01104] [TPS_SYST_01105] [TPS_SYST_01106] [TPS_SYST_02156] [TPS_SYST_02190] [TPS_SYST_02191] [TPS_SYST_02192] [TPS_SYST_02193]
[RS_SYST_00016]	Dedicated physical connections	[TPS_SYST_01043]
[RS_SYST_00017]	Mapping of signals to the same physical line	[TPS_SYST_01041]
[RS_SYST_00018]	Mapping of signals to different physical lines	[TPS_SYST_01044]
[RS_SYST_00019]	Mapping of signals to a specific physical line	[TPS_SYST_01043]
[RS_SYST_00020]	Exclusion of signals from a specific physical line	[TPS_SYST_01042]
[RS_SYST_00021]	ECU Communication via CAN	[TPS_SYST_01130]
[RS_SYST_00022]	ECU Communication via LIN	[TPS_SYST_01012] [TPS_SYST_01129] [TPS_SYST_02101] [TPS_SYST_02257] [TPS_SYST_05018] [TPS_SYST_05019]
[RS_SYST_00024]	ECU Communication via FlexRay	[TPS_SYST_01085] [TPS_SYST_01128]
[RS_SYST_00025]	Derivation of COM Stack Configuration Parameters from the System Template	[TPS_SYST_01030]
[RS_SYST_00027]	ECU Extract generation rules	[TPS_SYST_01000] [TPS_SYST_01002] [TPS_SYST_01003] [TPS_SYST_01016]
[RS_SYST_00028]	IPdu End-to-End Communication Protection support	[TPS_SYST_01070] [TPS_SYST_01071] [TPS_SYST_01072] [TPS_SYST_01073] [TPS_SYST_01074]
[RS_SYST_00029]	Dynamic length signals	[TPS_SYST_01065]
[RS_SYST_00031]	Distribution of Application and Vehicle Mode Requests	[TPS_SYST_01023]





Requirement	Description	Satisfied by
[RS_SYST_00033]	Software-to-ECU mapping variants	[TPS_SYST_01001]
[RS_SYST_00037]	Timing properties	[TPS_SYST_01075] [TPS_SYST_01076] [TPS_SYST_01077]
[RS_SYST_00038]	Support of SAE J1939 Protocol Features	[TPS_SYST_01106] [TPS_SYST_01132] [TPS_SYST_02107] [TPS_SYST_02108] [TPS_SYST_02109] [TPS_SYST_02190] [TPS_SYST_02191] [TPS_SYST_02192] [TPS_SYST_02193]
[RS_SYST_00039]	ECU Communication via Ethernet	[TPS_SYST_01086] [TPS_SYST_01088] [TPS_SYST_01089] [TPS_SYST_01090] [TPS_SYST_01091] [TPS_SYST_01094] [TPS_SYST_01095] [TPS_SYST_01096] [TPS_SYST_01097] [TPS_SYST_01098] [TPS_SYST_01108] [TPS_SYST_01131] [TPS_SYST_02156] [TPS_SYST_02217] [TPS_SYST_02218] [TPS_SYST_02219] [TPS_SYST_02220] [TPS_SYST_02221] [TPS_SYST_02222] [TPS_SYST_02223] [TPS_SYST_02224] [TPS_SYST_02225] [TPS_SYST_02226] [TPS_SYST_02227] [TPS_SYST_02228] [TPS_SYST_02229] [TPS_SYST_02232] [TPS_SYST_02233] [TPS_SYST_02234] [TPS_SYST_02235] [TPS_SYST_02236] [TPS_SYST_02237] [TPS_SYST_02238] [TPS_SYST_02239] [TPS_SYST_02240] [TPS_SYST_02241] [TPS_SYST_02242] [TPS_SYST_02243] [TPS_SYST_02244] [TPS_SYST_02245] [TPS_SYST_02247] [TPS_SYST_02248]
[RS_SYST_00042]	Support for Partial Networking	[TPS_SYST_01133] [TPS_SYST_02165] [TPS_SYST_02166] [TPS_SYST_02167]
[RS_SYST_00043]	Communication via Complex Drivers	[TPS_SYST_01115]
[RS_SYST_00044]	Description of custom bus systems	[TPS_SYST_01127]
[RS_SYST_00045]	Co-existing System artifacts in the same model	[TPS_SYST_03000]
[RS_SYST_00047]	Network and physical representation on signal level	[TPS_SYST_01062] [TPS_SYST_01063]
[RS_SYST_00048]	CAN with Flexible Data-Rate	[TPS_SYST_01154]
[RS_SYST_00049]	Support of Efficient COM for large data configuration	[TPS_SYST_02015] [TPS_SYST_02016] [TPS_SYST_02017] [TPS_SYST_02018] [TPS_SYST_02019] [TPS_SYST_02020] [TPS_SYST_02021] [TPS_SYST_02022] [TPS_SYST_02023] [TPS_SYST_02024] [TPS_SYST_02025] [TPS_SYST_02026] [TPS_SYST_02027] [TPS_SYST_02028] [TPS_SYST_02164] [TPS_SYST_03001]
[RS_SYST_00050]	Data transformation of inter-ECU communication	[TPS_SYST_02030] [TPS_SYST_02031] [TPS_SYST_02032] [TPS_SYST_02033] [TPS_SYST_02034] [TPS_SYST_02035] [TPS_SYST_02036] [TPS_SYST_02037] [TPS_SYST_02038] [TPS_SYST_02039] [TPS_SYST_02040] [TPS_SYST_02041] [TPS_SYST_02042] [TPS_SYST_02043] [TPS_SYST_02044] [TPS_SYST_02045] [TPS_SYST_02046] [TPS_SYST_02047] [TPS_SYST_02048] [TPS_SYST_02049] [TPS_SYST_02050] [TPS_SYST_02051] [TPS_SYST_02052] [TPS_SYST_02053] [TPS_SYST_02054] [TPS_SYST_02055] [TPS_SYST_02056] [TPS_SYST_02057] [TPS_SYST_02074] [TPS_SYST_02075] [TPS_SYST_02080] [TPS_SYST_02092] [TPS_SYST_02093] [TPS_SYST_02094] [TPS_SYST_02121] [TPS_SYST_02123] [TPS_SYST_02124] [TPS_SYST_02125] [TPS_SYST_02126] [TPS_SYST_02127] [TPS_SYST_02128] [TPS_SYST_02129] [TPS_SYST_02130] [TPS_SYST_02131] [TPS_SYST_02132] [TPS_SYST_02156] [TPS_SYST_02195] [TPS_SYST_02212] [TPS_SYST_02213] [TPS_SYST_02214]
[RS_SYST_00051]	Support of COM Based Data Transformation	[TPS_SYST_02058]





Requirement	Description	Satisfied by
[RS_SYST_00052]	Ethernet Switch Configuration	[TPS_SYST_03002] [TPS_SYST_03003] [TPS_SYST_03004] [TPS_SYST_03005] [TPS_SYST_03006] [TPS_SYST_03007] [TPS_SYST_03008] [TPS_SYST_03009] [TPS_SYST_03010] [TPS_SYST_03011] [TPS_SYST_03013]
[RS_SYST_00053]	The System Template shall provide the ability to define naming conventions for public symbols	[TPS_SYST_05015]
[RS_SYST_00054]	Support of Secured Pdus	[TPS_SYST_02059] [TPS_SYST_02060] [TPS_SYST_02148] [TPS_SYST_02149] [TPS_SYST_02152] [TPS_SYST_02153] [TPS_SYST_02154] [TPS_SYST_02171] [TPS_SYST_02172] [TPS_SYST_02173] [TPS_SYST_02189] [TPS_SYST_05020] [TPS_SYST_05021] [TPS_SYST_05022] [TPS_SYST_05023] [TPS_SYST_05024] [TPS_SYST_05025] [TPS_SYST_05026] [TPS_SYST_05027] [TPS_SYST_05028]
[RS_SYST_00055]	Support of Container Pdus	[TPS_SYST_01056] [TPS_SYST_02061] [TPS_SYST_02062] [TPS_SYST_02063] [TPS_SYST_02064] [TPS_SYST_02065] [TPS_SYST_02066] [TPS_SYST_02097] [TPS_SYST_02098] [TPS_SYST_02099] [TPS_SYST_02100] [TPS_SYST_02196] [TPS_SYST_03014]
[RS_SYST_00056]	E2E-protected communication	[TPS_SYST_02067] [TPS_SYST_02068] [TPS_SYST_02069] [TPS_SYST_02070] [TPS_SYST_02071] [TPS_SYST_02072] [TPS_SYST_02073] [TPS_SYST_02134] [TPS_SYST_02135] [TPS_SYST_02155]
[RS_SYST_00057]	Assigning communication graphs to particular RTE Implementation Plug-Ins	[TPS_SYST_02197]
[RS_SYST_00058]	The System Template shall support the usage of the TLV encoding in SOME/IP messages	[TPS_SYST_02211] [TPS_SYST_05016] [TPS_SYST_05017]
[RS_SYST_00059]	The System Template shall support the translation between signal-based and service-oriented communication.	[TPS_SYST_03022] [TPS_SYST_03023] [TPS_SYST_03024] [TPS_SYST_03025] [TPS_SYST_03026] [TPS_SYST_03027] [TPS_SYST_03028] [TPS_SYST_03029] [TPS_SYST_03030] [TPS_SYST_03031] [TPS_SYST_03032] [TPS_SYST_03033] [TPS_SYST_03034] [TPS_SYST_03036] [TPS_SYST_03037] [TPS_SYST_03038] [TPS_SYST_03039] [TPS_SYST_03040] [TPS_SYST_03041] [TPS_SYST_03042] [TPS_SYST_03043] [TPS_SYST_03044] [TPS_SYST_03045] [TPS_SYST_03046] [TPS_SYST_03047] [TPS_SYST_03048] [TPS_SYST_03049] [TPS_SYST_03051]

Table 1.2: Requirements Tracing

1.3 Requirements not fulfilled by TPS requirements

This section contains a list of requirements that are not yet fulfilled by TPS requirements.

Requirement	Description	Satisfied by
[RS_SYST_00015] Bus bandwidth	The System Template shall support bandwidth calculation as a constraint for the definition of the Communication Matrix.	chapter Topology (3); Communication (chapter 6)
[RS_SYST_00023] ECU Communication via MOST	The System Template has to cover the system communication via MOST.	not covered

<p>[RS_SYST_00025] Derivation of ECU Configuration Parameters from the System Template</p>	<p>The System Template shall enable the configuration of the Com Stack of the ECU. It handles those parameters that are necessary to describe the inter-ECU communication. Configuration parameters local to an ECU are not in the scope of the System Template.</p>	<p>Harmonization between Upstream Templates and ECU Configuration (chapter C)</p>
<p>[RS_SYST_00026] Fibex compatibility</p>	<p>Whenever there is a considerable overlap between the System Template and the ASAM FIBEX Standard, the System Template shall adopt the structures of the ASAM FIBEX Standard.</p>	<p>AUTOSAR System Template and ASAM FIBEX (chapter 1.8)</p>
<p>[RS_SYST_00032] Topology Variants</p>	<p>The System Template shall provide the means to describe topology variants with optional/alternative ECUs and communication clusters.</p>	<p>chapter Variant Handling 1.7.2 and chapter Topology 3.</p>
<p>[RS_SYST_00033] Software-to-ECU mapping variants</p>	<p>The System Template shall provide the means to describe alternative mappings of software components to ECUs.</p>	<p>chapter 1.7.2 Variant Handling and chapter 5.1 Software Component Mapping.</p>
<p>[RS_SYST_00034] Timing variants</p>	<p>The System Template shall provide the means to describe alternative timing properties (e.g. trigger type, period, priority) and timing constraints (e.g. latency, age).</p>	<p>chapter 1.7.2 Variant Handling and chapter 6 Communication.</p>
<p>[RS_SYST_00035] Data mapping variants</p>	<p>The System Template shall provide the means to describe data mapping Variants.</p>	<p>chapter 1.7.2 Variant Handling and chapter 5.2 Data Mapping.</p>
<p>[RS_SYST_00036] Communication variants</p>	<p>The System Template shall provide the means to describe communication variants, such as alternative signal-to-PDU mappings, alternative communication paths, and alternative signal and PDU properties (e.g. data type, data length).</p>	<p>chapter 1.7.2 Variant Handling and chapter 6 Communication.</p>
<p>[RS_SYST_00040] Timing constraints</p>	<p>The System Template shall provide the means to describe the timing constraints of a system's dynamics, which are determined by the consumption of computation, communication, and other hardware resources.</p>	<p>Timing Extensions (chapter 1.7.3)</p>
<p>[RS_SYST_00041] Variants in ECU Extract</p>	<p>The ECU Extract shall support variability of elements taken over or derived during the transformation from the System Description.</p>	<p>Variant Handling in ECU Extract (chapter 12.6)</p>

1.4 Methodology for Defining Formal Template

Figure 1.1 illustrates the overall methodology used to define formal templates. As is explained in the "Generic Structure Template" [2], it is important to separate a precise and concise model of the information that needs to be captured from the concrete XML-DTDs, XML-Schemas or other technology that is used to define the actual templates.

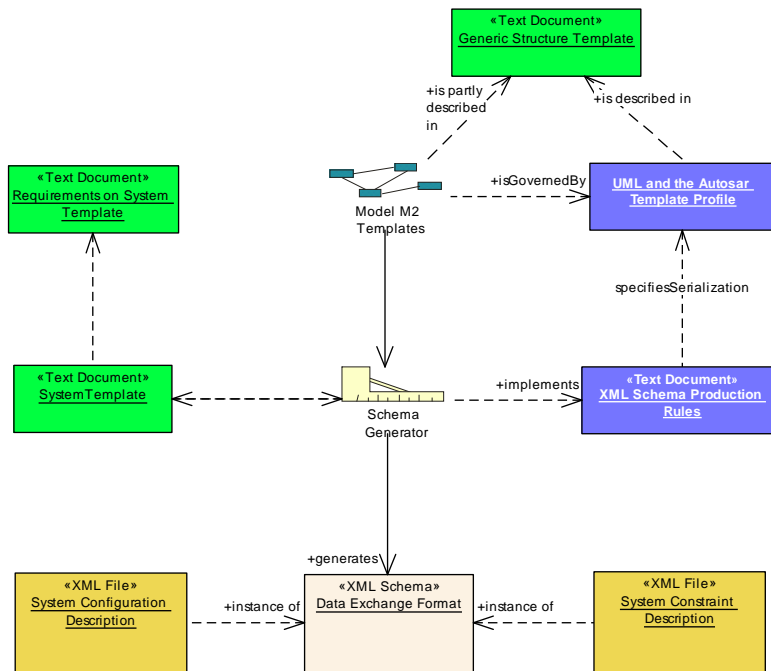


Figure 1.1: Methodology to define templates in AUTOSAR

The following documents describe the various aspects of the methodology:

1. The document called `System Template` (this document) describes the information that can be captured in the "system constraint" and "system configuration" description, independently from the mapping of this model on XML-technology. This document is based upon the AUTOSAR meta-model and contains an elaborate description of the semantics (the precise meaning) of all the information that can be captured within the relevant parts of this meta-model.
2. The `UML and the AUTOSAR Template Profile` [2] describes the basic concepts that should be used when creating content of the meta-model.
3. The document called "XML Schema Production Rules" [3] describes how XML is used and how the meta-model designed in the "System Template" should be translated by the "Schema Generator" (MMT) into XML-Schema (XSD) "Data Exchange Format". This "formalization strategy" is to be used for all data that is formally described in the meta-model. In particular this document is worth to read in order to understand the mapping of the meta-model and the XML based System template.

4. The "Generic Structure Template" [2] describes the top level structure which is common to all AUTOSAR templates and provides AUTOSAR standard mechanisms of modeling elements and patterns.
5. The concrete "Template", the "Data Exchange Format" is an XML schema which is generated out of the meta-model described in the "System Template" using the approach and the patterns defined in the "XML Schema Production Rules". This schema is typically used as input to tools. The M1-level system descriptions are XML files which can be validated against the schema. In that sense they are instances of the schema defining the XML representation of the template.

1.5 Scope

This document describes the system template and its use for the System Constraint Description and the System Configuration Description. In general a filled system template defines the relationship between the pure Software View on the System (represented by a top level SW Component Composition) and a Physical System Architecture with networked ECU instances. The system template is used in two stages of the "AUTOSAR Methodology" [4] (see Figure 1.2).

- As System Constraint Description it serves as input to the AUTOSAR system generator
- As System Configuration Description it defines the output of the AUTOSAR System Configuration Generator and serves as input to the AUTOSAR ECU Configuration Generator for the different ECUs defined in the description.
- As ECU Extract of the System Configuration Description it describes the ECU specific view on the System Description. It is individually generated for each of the System's ECU as the output of the AUTOSAR ECU Configuration Generator.

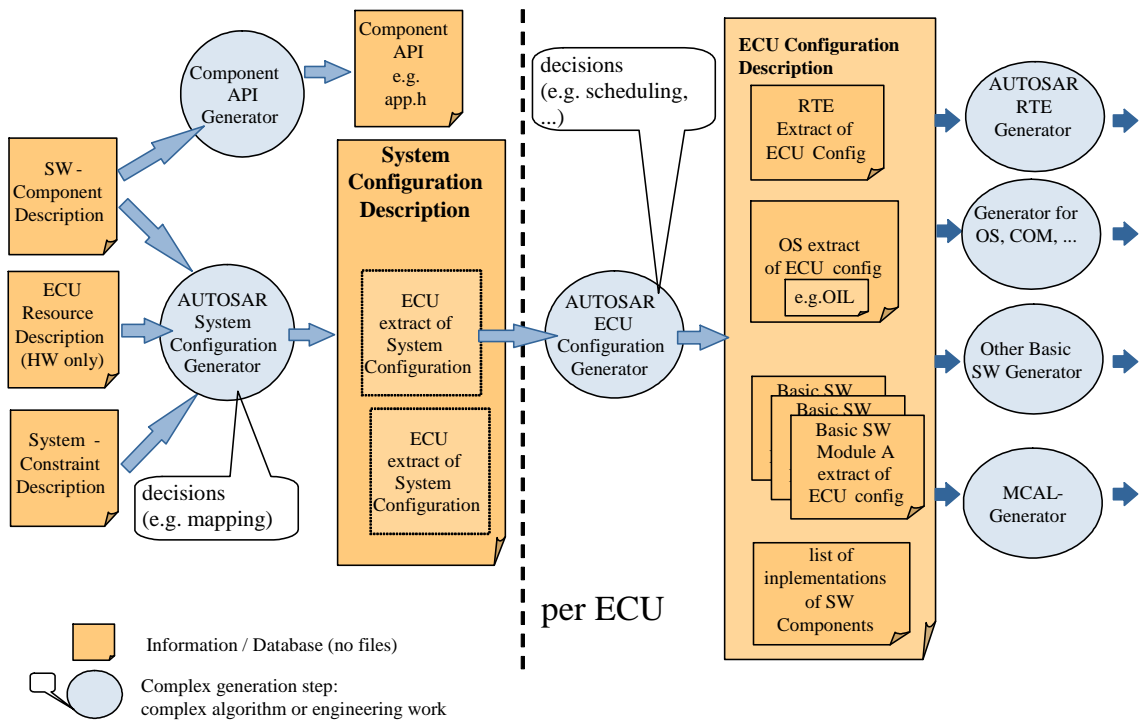


Figure 1.2: AUTOSAR Methodology

The System Template defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints, which will be defined in detail in the following chapters. Figure 1.3 gives an overview how these are used in the two different descriptions.

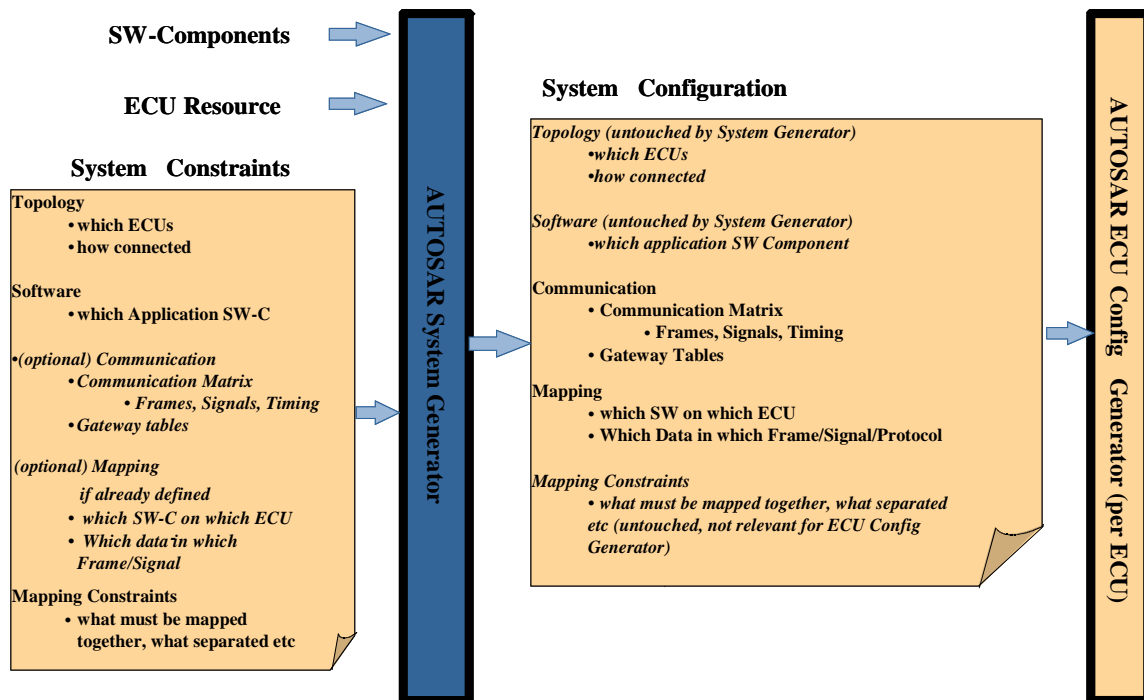


Figure 1.3: Scope of System Constraint Description and System Configuration Description

On Figure 1.3 some of the elements are marked *optional* for the System Constraint Description. If one starts with a new AUTOSAR project, these elements may not be present in the System Constraint Description. No (at least partial) functionality has been mapped yet, thus the communication matrix is not populated. But in most cases, many functional mappings are already predefined and contribute to the population of the communication matrix with their associated signals, thus being present in the System Constraint Description.

Reasons for such a predefinition are manifold. In some cases, hardware setup dictates where certain functionality resides, in some cases, a partial or complete communication matrix and/or completely configured ECUs (HW and SW) of another system (vehicle) has to be taken over. This approach is eased by the fact that System Configuration and System Constraint Description use the same format. That way it is possible to reuse parts of a System Configuration Description of the other system/vehicle in the actual System Constraint Description.

Furthermore, in the figure some of the elements are marked *untouched* for the System Configuration Description. This can have two reasons:

- The System Generator does not modify neither the Topology (networked ECUs) nor the Software, so these parts are just moved from System Constraint Description to System Configuration Description during the generation step.
- In a completed System Configuration Description, all SW components and all ECU-to-ECU communication have been mapped. Thus mapping constraints that limit the flexibility in the mapping phase of the system generator are obsolete

and will not be used in subsequent generator steps. They may however still be present for documentation and validation reasons.

Even if the communication matrix is determined as the result of the system configuration, the ECUs still have to be configured. This is done by the ECU configuration generator, which takes the System Configuration description as input and generates the ECU configuration description. The following guiding principles have been used to determine which information shall be part of the System Configuration Description and which goes into the ECU Configuration Description:

- Information that is common for several ECUs and has to be agreed, shall be part of the System Configuration Description and is thus covered by the System Template.
- Information, that only has ECU-local relevance is part of the ECU Configuration Description.

Thus the ECU Configuration Description will include the OS-schedule, the RTE-configuration and last but not least the configuration of the ECU basic software including the concrete communication drivers on that ECU.

1.6 UML Meta-Model

This chapter gives an overview of the AUTOSAR Unified Modeling Language (UML) meta-model. All AUTOSAR templates use a common meta-model. The templates describe software components, ECU resources, the Basic Software Modules, the ECU Configuration Parameters (ECU Configuration Description and ECU Configuration Parameter Definition) and the System.

The System Template defines all elements, their parameters and their relations, which are necessary for the System Constraint Description and the System Configuration Description.

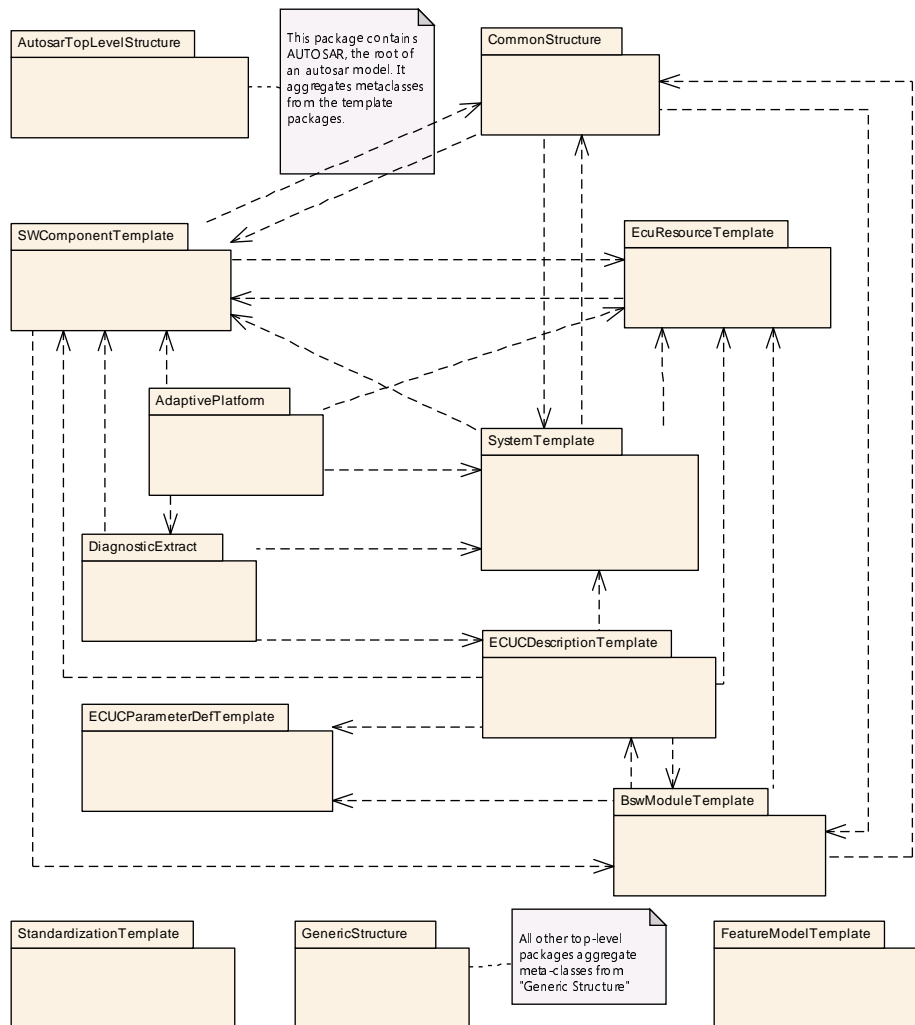


Figure 1.4: AUTOSAR Package Overview

Figure 1.4 shows the overall structure of the meta-model.

The dashed arrows in the diagram describe dependencies in terms of import-relationships between the packages within the meta-model. For example, the package `SystemTemplate` imports meta-classes defined in the packages `GenericStructure` [2], `SWComponentTemplate` [5] and `ECUResourceTemplate` [6].

For clarification, please note that the package `GenericStructure` contains some fundamental infrastructure meta-classes and common patterns that are described in [2]. As these are used by all other template specification the dependency associations are not depicted in the diagram for the sake of clarity.

Generic Structure provides details about

- Autosar Top level structure,
- Commonly used metaclasses and primitives
- Variant Handling

- Documentation

The ECU Resource Template deals with the description of the hardware resources of an ECU. The collection of all ECUs, which are integrated in the car, are described in the topology part of the System Configuration Description/System Constraint Description. Each of these ECUInstances uses the ECU Resource Template to describe the hardware resources. That's the reason, why the topology part has references to the ECU Resource Description.

The SW component description describes the SW components as well as their communication by data elements. The top-level software composition ([RootSwCompositionPrototype](#)) is part of the System Template (Software). This top-level software composition contains the functionality of the full system and describes the complete application software architecture of this system. The definition of the top level software composition uses the elements defined in the SW Component Template, like e.g. [SwComponentType](#), [PortInterface](#), [AssemblySwConnector](#) and [DelegationSwConnector](#). That's why the System Description has references to the Software Component Description. The top level software composition is described in more detail in chapter 4.

Every template starts with an element `AUTOSAR`. While the models created in accordance to this guide are independent of the used formalization, it may still help the reader's understanding to note that `AUTOSAR` would also typically be the root element of a XML Schema generated from such a model. `AUTOSAR` can then contain one or more nested packages, simply allowing to further structure the contents of the M1 model.

1.7 Document Conventions

Technical terms are typeset in mono spaced font, e.g. `PortPrototype`. As a general rule, plural forms of technical terms are created by adding "s" to the singular form, e.g. `PortPrototypes`. By this means the document resembles terminology used in the AUTOSAR XML Schema.

This document contains constraints in textual form that are distinguished from the rest of the text by a unique numerical constraint ID, a headline, and the actual constraint text starting after the `[` character and terminated by the `]` character.

The purpose of these constraints is to literally constrain the interpretation of the AUTOSAR meta-model such that it is possible to detect violations of the standardized behavior implemented in an instance of the meta-model (i.e. on M1 level).

Makers of AUTOSAR tools are encouraged to add the numerical ID of a constraint that corresponds to an M1 modeling issue as part of the diagnostic message issued by the tool.

The attributes of the classes introduced in this document are listed in form of class tables. They have the form shown in the example of the top-level element `AUTOSAR`:

Class	AUTOSAR			
Package	M2::AUTOSARTemplates::AutosarTopLevelStructure			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. Tags: xml.globalElement=true			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. Tags: xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. Stereotypes: atpStructuredComment Tags: xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. Tags: xml.sequenceOffset=20

Table 1.3: AUTOSAR

The first rows in the table have the following meaning:

Class: The name of the class as defined in the UML model.

Package: The UML package the class is defined in. This is only listed to help locating the class in the overall meta model.

Note: The comment the modeler gave for the class (class note). Stereotypes and UML tags of the class are also denoted here.

Base Classes: If applicable, the list of direct base classes.

The headers in the table have the following meaning:

Attribute: The name of an attribute of the class. Note that AUTOSAR does not distinguish between class attributes and owned association ends.

Type: The type of an attribute of the class.

Mul.: The assigned multiplicity of the attribute, i.e. how many instances of the given data type are associated with the attribute.

Kind: Specifies, whether the attribute is aggregated in the class (*aggr* aggregation), an UML attribute in the class (*attr* primitive attribute), or just referenced by it (*ref* reference). Instance references are also indicated (*iref* instance reference) in this field.

Note: The comment the modeler gave for the class attribute (role note). Stereotypes and UML tags of the class are also denoted here.

Please note that the chapters that start with a letter instead of a numerical value represent the appendix of the document. The purpose of the appendix is to support the explanation of certain aspects of the document and does not represent binding conventions of the standard. The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([7]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([7]).

1.7.1 Detailed Representation of InstanceRef Associations

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [2]. Each "instanceRef" association can both be represented by the short form and by an detailed representation. For readability the diagrams in the main body of the specification use the short form. The detailed descriptions can be found in the Appendix B.

1.7.2 Variant Handling

The System Template supports the creation of Variants in many of its model elements. In the Metamodel all locations that may exhibit variability are marked with the stereotype `atpVariation`. This allows the definition of possible variation points. Tagged Values are used to specify additional informations.

There are four types of locations in the metamodel which may exhibit variability:

- Aggregations
- Associations
- Attribute Values
- Classes providing property sets

The reasons for the attachment of the stereotype `atpVariation` to certain model elements and the consequences for other model elements are explained in class tables in the following chapters. More details about the AUTOSAR Variant Handling Concept can be found in the AUTOSAR Generic Structure Template [2].

1.7.3 Timing Extensions

With AUTOSAR Release 4.0 a new set of concepts for the description and analysis of end-to-end timing constraints is introduced by the Specification of Timing Extensions. A subset of these extensions aims for the system level and can be used to enhance the descriptions that are already available in the System Template.

A dedicated description of the timing extensions that can be used at system level is given in chapter 3 (System timing) in the Specification of Timing Extensions [8].

1.7.4 Documentation Support

With AUTOSAR Release 4.0 the AUTOSAR XML schema provides support for integrated and well structured documentation. More details about the AUTOSAR Documentation Support concept can be found in the AUTOSAR Generic Structure Tem-

plate [2]. An optional documentation block can be applied to any identifiable element. Furthermore, as shown in figure 1.5, the System Template provides the possibility of adding additional documentation to several non-identifiable elements. The documentation of a *System* is composed of several chapters.

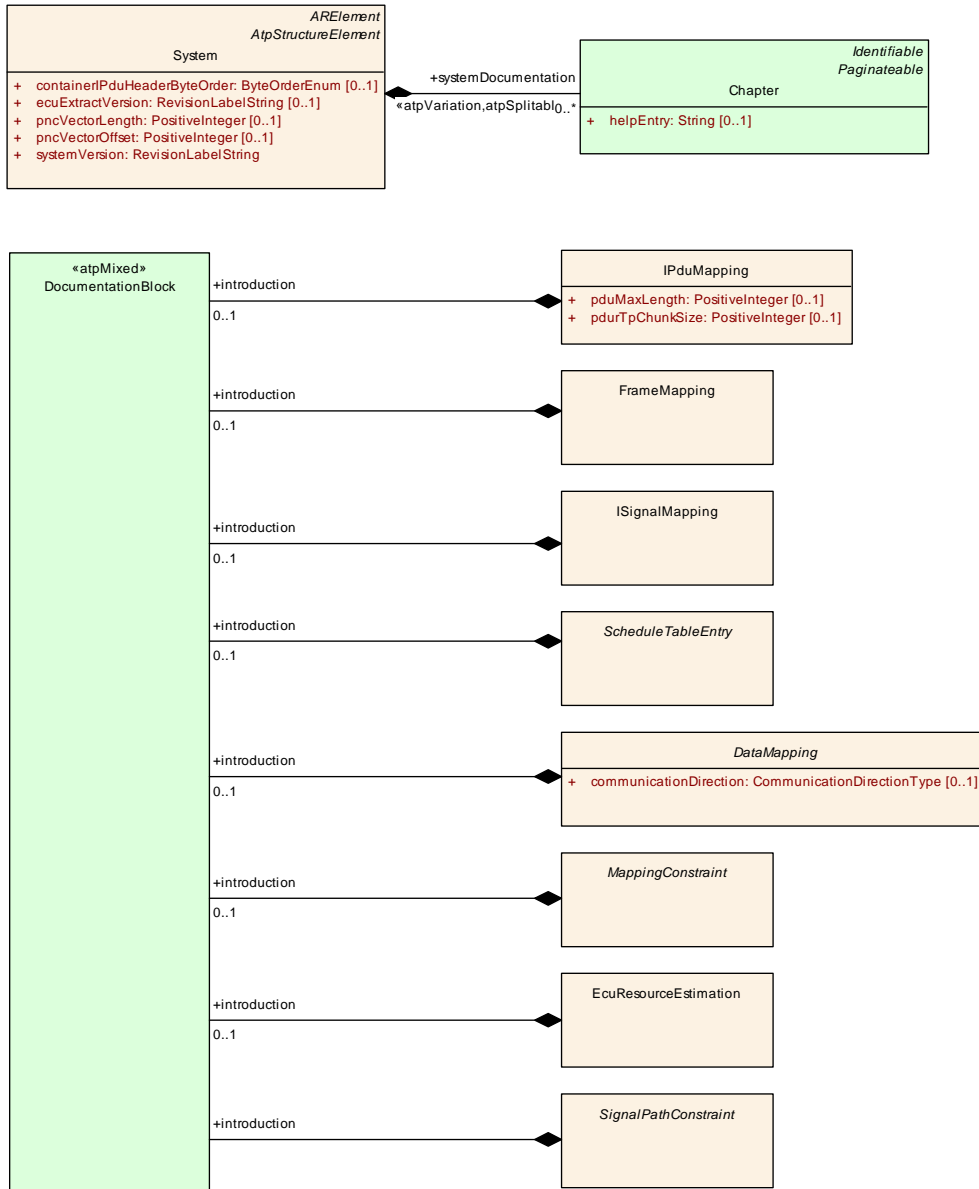


Figure 1.5: System Template Documentation Support

1.7.5 Stereotype `atpSplitable` in the System Template

The stereotype `<<atpSplitable>>` is used in the System Template to support step-wise processes, where the System Configuration Description is completed incrementally over a development process. Example:

1) Description of Communication only consists of interaction signals (ISignal). This is enough information to create an individual ECU's RTE, and even contains enough information to configure an ECU where the actual Frame/Pdu communication is being handled post-build.

2) In a second step, the communication matrix is being completed for a concrete vehicle. Pdus and Frames, along with their Triggerings are being added to the previous System Description. This model then contains the full information about an ECU's communication, especially containing the additional information to generate the post build information.

So, in this 2-step approach, an OEM could deliver the incomplete ECU extract from step (1) to the ECU integrator, who can then build a complete software image for the ECU. In the 2nd step, the ECU extract will be completed by the previously missing information, but as the first extract will still be valid due to the `<<atpSplitable>>` construct, the ECU including the flashed image from step (1) can be (re)used as it is, and just will be completed with the post build information, e.g. Frames and Pdus.

Further details about the `<<atpSplitable>>` stereotype can be found in the Generic Structure Template [2].

1.8 AUTOSAR System Template and ASAM FIBEX

FIBEX (Field Bus Exchange Format) [9] is an XML exchange format proposed for data exchange between tools that deal with bus communication Systems. The format supports the most common automotive data buses: LIN [10], CAN [11], MOST [12], FlexRay [13]. The covered areas of the exchange format are the functional network, system topology and the communication level. The functional network describes the software architecture of the system. In the system topology the logical layout of the system is described. This means it is documented which ECU is connected to which bus. The central purpose of a communication system is the exchange of frames with certain properties. The format is able to describe frames and their timing properties.

In future versions of the System Template a common subset between ASAM Fibex and Autosar will be harmonized. The current version of the System Template contains already the ASAM FIBEX description for communication and topology. Due to requirements of AUTOSAR some extensions were made to those descriptions. For instance the communication part is extended by a concept for PDUs (I-Pdus and N-Pdus). The harmonization between ASAM Fibex and AUTOSAR System Template is not finalized at this time.

In the UML Meta-Model the FIBEX contents are located in an own FIBEX UML Package. The top level `FibexElement` is referenced by the top level element `System` of the System Template. Similar to the usage of the `ARElement`, specializations of the `FibexElement` represent elementary building blocks within the FIBEX package. Each of this elements will be described in more detail in the following chapters.

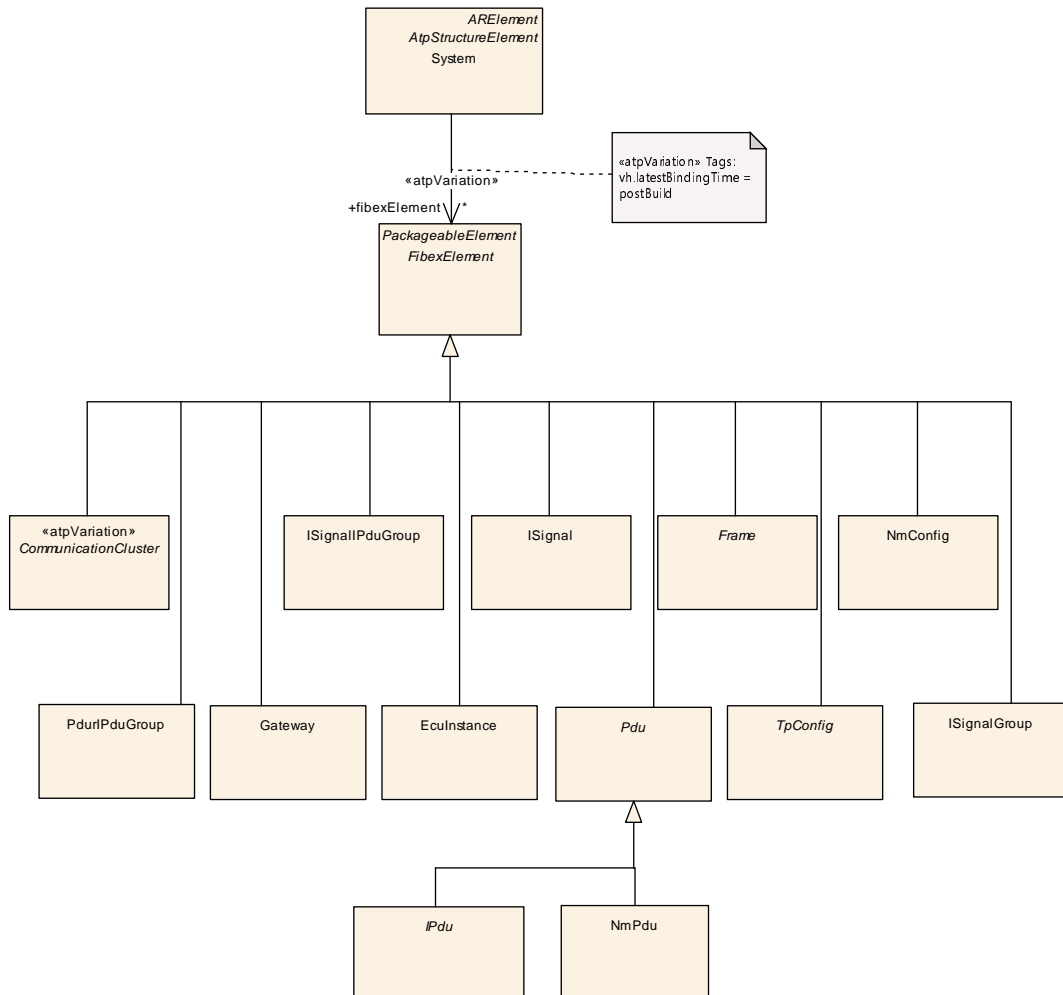


Figure 1.6: Fibex Elements

2 System

The top level element of the System Template is the class *System*, as shown in figure 2.1.

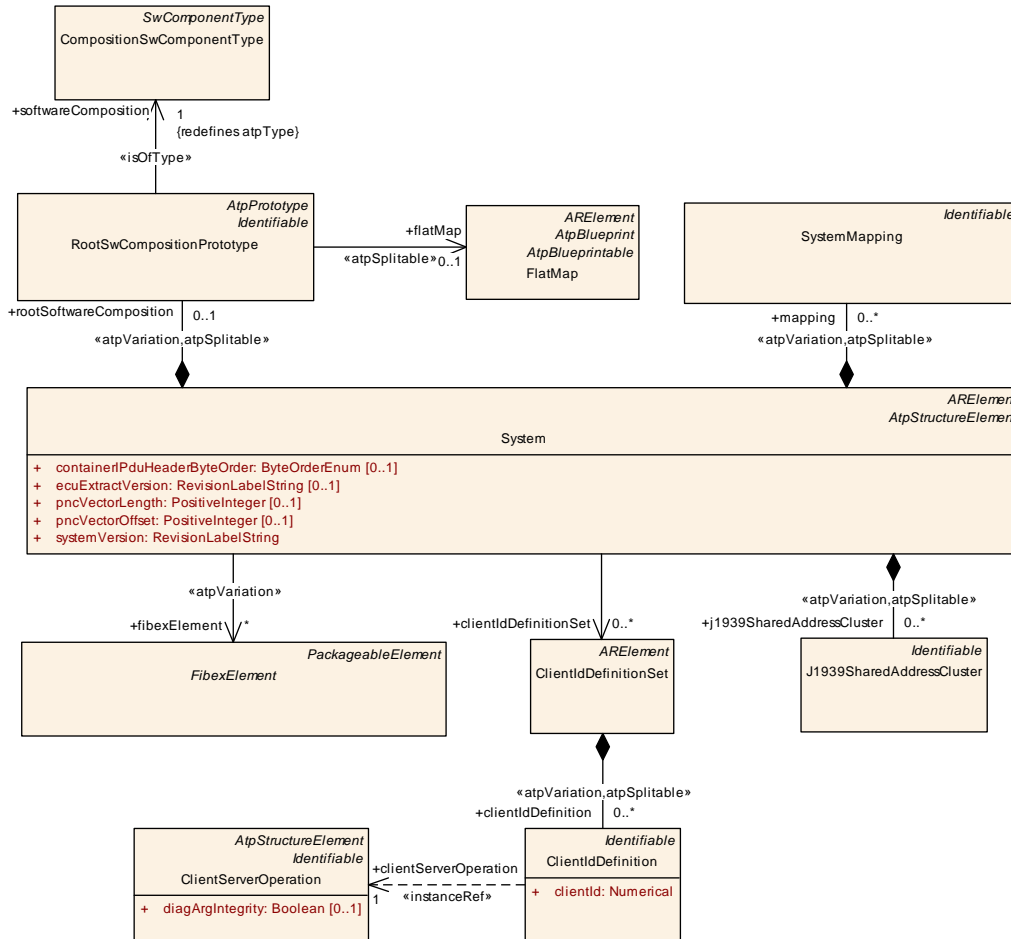


Figure 2.1: System Template Overview

Class	System			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	<p>The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints.</p> <p>The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology.</p> <p>Tags:atp.recommendedPackage=Systems</p>			
Base	ARElement , ARObject , AtpClassifier , AtpFeature , AtpStructureElement , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
clientIdDefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System.





Class	System			
containerIPduHeaderByteOrder	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus.
ecuExtractVersion	RevisionLabelString	0..1	attr	Version number of the Ecu Extract.
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology. All Fibex Elements used within a System Description shall be referenced from the System Element. atpVariation: In order to describe a product-line, all Fibex Elements can be optional. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
j1939SharedAddressCluster	J1939SharedAddressCluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
mapping	SystemMapping	*	aggr	Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints). In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping. This element is not required when the System description is used for a network-only use-case. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
pncVectorLength	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftwareComposition	RootSwCompositionPrototype	0..1	aggr	Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case. atpVariation: The RootSwCompositionPrototype can vary. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime





Class	System			
systemDocumentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table 2.1: System

[System](#) has relationships to all elements that define a system constraint description or system configuration description. It aggregates the [SystemMapping](#) and [RootSwCompositionPrototype](#) elements. [SystemMapping](#) deals with mapping of software components to ECUs as well as with the mapping of data elements that are to be exchanged between software components onto signals and frames. The [RootSwCompositionPrototype](#) element contains a reference to the top level software composition.

[constr_3028] FibexElements [Each [FibexElement](#) that is used in the System Description shall be referenced by the [System](#) element in the role [FibexElement](#).]()

[FibexElements](#) can be defined in a stand alone and reusable way (hence they can simply be created in any package like ARElements), but on the other hand it shall be clear that a certain [FibexElement](#) actually belongs to a certain System Description. Thus, all [FibexElements](#) used within a System Description (i.e. contributing to the specification of the System communication and topology) shall be referenced from the [System](#) element. More details about the integration of FIBEX into the System Template will be given in chapter 1.8.

[TPS_SYST_01002] System Category [The [System](#) shall have a `category` element defined which indicates the role of this work product.]([RS_SYST_00003](#), [RS_SYST_00027](#))

[TPS_SYST_01003] Standardized System Category Definitions [The standardized System category definitions are defined in [Table 2.2](#).]([RS_SYST_00003](#), [RS_SYST_00027](#))

category	Meaning
SYSTEM_CONSTRAINTS	The System class is used to describe System Constraints. In this usage, it forms the core element of a System Constraints Description, serving as an input to the AUTOSAR System Generator.
SYSTEM_DESCRIPTION	The System class is used to describe the System Configuration of a complete AUTOSAR System. In this usage, it forms the core element of a System Description, the output of the AUTOSAR System Generator.
SYSTEM_EXTRACT	The System class is used to describe a subsystem specific view on the complete System Description. The System Extract is not fully decomposed and still contains compositions. The SYSTEM_EXTRACT is the basis for designing subsystems.

<i>category</i>	<i>Meaning</i>
ECU_EXTRACT	The System class is used to describe the ECU specific view on the complete System Description. In this usage, it forms the core element of ECU Extract, the output of the AUTOSAR ECU Configuration Extractor. The ECU Extract is fully decomposed and contains only atomic software components. The ECU Extract is the basis for setting up the ECU Configuration.
ABSTRACT_SYSTEM_DESCRIPTION	This System is used to describe a functional (solution-independent/abstract) system design. It can be taken as basis for the development of the SYSTEM_DESCRIPTION. No structural constraints are applied on the transformation of the ABSTRACT_SYSTEM_DESCRIPTION to the SYSTEM_DESCRIPTION.
ECU_SYSTEM_DESCRIPTION	This System is used to describe the closed view on one ECU (note that an AUTOSAR ECU is defined being one microprocessor running one AUTOSAR Stack). It can be derived from a SYSTEM_EXTRACT or it can be designed independently and mapped to a SYSTEM_EXTRACT. The ECU_SYSTEM_DESCRIPTION is not fully decomposed and still may contain compositions.
RPT_SYSTEM	System which describes the rapid prototyping algorithm in the format of AUTOSAR Software Components. For more details see the Software Component Template [5] and TR_Methodology [4].

Table 2.2: System class categories

Note: SYSTEM_EXTRACT does not prescribe the number of micro controllers / cores for one ECU from the OEM perspective.

- Supplier decides to design one AUTOSAR ECU with multicore support leads to one ECU_EXTRACT supporting one AUTOSAR stack
- Supplier decides to design two AUTOSAR ECUs (i.e., two micro-controllers) in one box leads to two ECU_EXTRACTs supporting two AUTOSAR stacks

[constr_3027] Existence of `ecuExtractVersion` [In case the category of the System is SYSTEM_EXTRACT or ECU_EXTRACT the `ecuExtractVersion` attribute shall be defined.]()

2.1 ClientIdDefinitionSet

In the [ClientIdDefinitionSet](#) all Client Identifiers of the transaction handle used for a inter-ECU client server communication can be defined that belong to the [System](#) that refers the [ClientIdDefinitionSet](#).

Class	ClientIdDefinitionSet
Package	M2::AUTOSARTemplates::SystemTemplate
Note	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. Tags: atp.recommendedPackage=ClientIdDefinitionSets





Class	ClientIdDefinitionSet			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
clientId Definition	ClientIdDefinition	*	aggr	Definition of a Client Identifier that will be used by the RTE in a inter-ECU client-server communication. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 2.3: ClientIdDefinitionSet

Class	ClientIdDefinition			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	Several clients in one client-ECU can communicate via inter-ECU client-server communication with a server on a different ECU, if a client identifier is used to distinguish the different clients. The Client Identifier of the transaction handle that is used by the RTE can be defined by this element.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
clientId	Numerical	1	attr	The Client Identifier of the transaction handle used for an inter-ECU client server communication is defined by this attribute. If defined the RTE generator shall use this client Id.
clientServer Operation	ClientServerOperation	1	iref	Reference to the ClientServerOperation that is called by the client.

Table 2.4: ClientIdDefinition

[constr_3117] Allowed value of attribute `clientId` [Within the context of one `ClientIdDefinition`, the value of attribute `clientId` shall be in the range of `ClientIdRange.lowerLimit` and `ClientIdRange.upperLimit` for the `ClientIdRange` that is aggregated by the `EcuInstance` onto which the `SwComponentPrototypes` included in the `ClientIdDefinition.clientServerOperation` are mapped.]()

Please note that the `clientId` is bound to the ClientServer relationship and does not represent a globally unique identifier of the Client call. ClientIds can be reused in the context of a different ClientServer relationship.

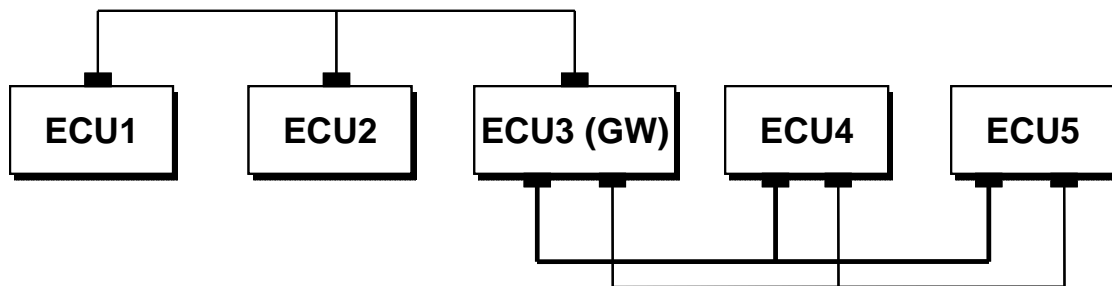
[constr_3118] Valid reference target for `ClientIdDefinition.clientServerOperation.contextPort` [In the context of the definition of a `ClientIdDefinition`, the reference `clientServerOperation.contextPort` shall only refer to an `RPortPrototype`.]()

Rationale: the definition of a client ID does only make sense in the context of a client of a `ClientServerOperation`.

3 Topology

This chapter explains how a vehicle’s physical System Topology is being modeled in AUTOSAR (Example: Figure 3.1). A topology is formed by a number of `EcuInstances` that are interconnected to each other in order to form ensembles of ECUs and `CommunicationClusters`, which are further detailed by providing information on bus-specific properties.

CAN CommunicationCluster:
1 PhysicalChannel



Redundant FlexRay CommunicationCluster:
2 PhysicalChannels (bold line, thin line)

Figure 3.1: Example for a Communication Cluster within a physical network topology

In the AUTOSAR methodology [4] the topology description is one of the inputs for the System Generator. It serves as constraints for mapping the Software Components (see chapter 5.1) contained in the `RootSwCompositionPrototype` as well as for defining the System Communication matrix (see chapter 6). Gateways which allow the exchange of Signals between `CommunicationClusters` are covered in chapter 8.

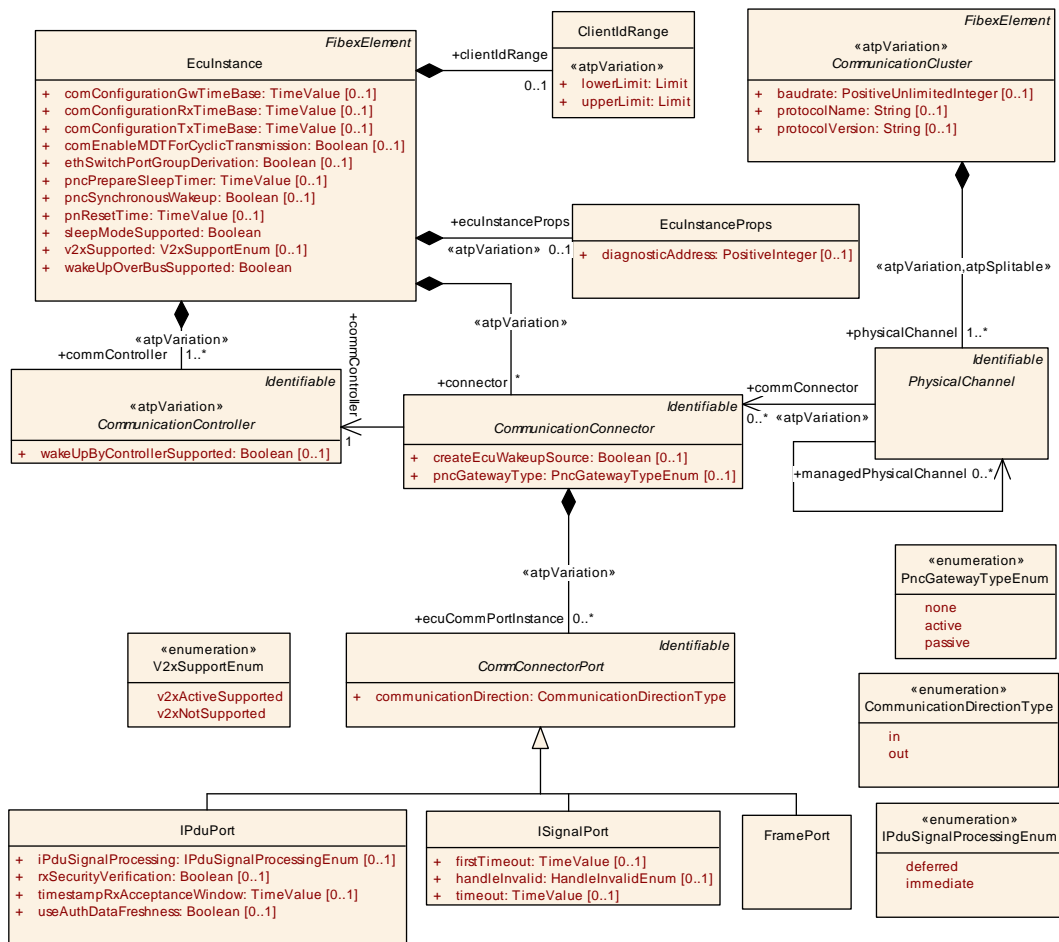


Figure 3.2: Topology elements (Topology)

3.1 ECUs and their communication capabilities

Within a System Topology, the ECUs actually being connected with each other are described in the form of `EcuInstance`s. An `EcuInstance` needs to have one or more `CommunicationController`, the actual hardware device by means of which devices send and receive frames from the communication medium. Furthermore, the `EcuInstance` has one or more `CommunicationConnectors` which describe the bus interfaces of the ECUs and to specify the sending/receiving behavior.

[TPS_SYST_01004] Definition of AUTOSAR ECU [In the AUTOSAR sense an ECU means a microcontroller plus peripherals and the according software/configuration. Therefore, each microcontroller requires its own ECU Configuration.] ([RS_SYST_00013](#))

3.1.1 ECU Instance

[TPS_SYST_01005] Definition of `EcuInstance` [`EcuInstance` describes the presence of a microcontroller in the vehicle. Within an `EcuInstance` class only those properties are described that are subject to system configuration.] (*RS_SYST_00013*)

The actual description of the ECU hardware resources is done by the means of the ECU Resource Template [6]: It uses the `HwElement` class and its aggregated hardware elements for defining a specific ECU type.

[TPS_SYST_01006] Assign ECU type to `EcuInstance` [The process of assigning an ECU type to `EcuInstance` is a mapping step (see [TPS_SYST_01019]) and performed latest in the System Generation step.] (*RS_SYST_00013*)

An `EcuInstance` can serve as a gateway if it is connected to two or more different clusters by two or more of its `CommunicationControllers`.

Class	EcuInstance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description. Tags: atp.recommendedPackage=EcuInstances			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
associatedComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignalIPduGroups are applicable for which Communication Connector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPduGroups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup.
associatedConsumedProvidedServiceInstanceGroup	ConsumedProvidedServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
associatedPdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PdurIPdu Groups are applicable for which Communication Connector/ ECU.
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication.
comConfigurationGwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRouteSignals of the AUTOSAR COM module in seconds.
comConfigurationRxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRx of the AUTOSAR COM module in seconds.





Class	EcuInstance			
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds.
comEnable MDTForCyclic Transmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclic Timing assigned or eventControlledTiming with numberOfRepetitions > 0).
commController	Communication Controller	1..*	aggr	CommunicationControllers of the ECU. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
dltConfig	DltConfig	0..1	aggr	Describes the Dlt configuration on this EcuInstance.
dolpConfig	DolpConfig	0..1	aggr	Dolp configuration on this EcuInstance. Tags: atp.Status=draft
ecuInstance Props	EcuInstanceProps	0..1	aggr	Additional properties of the EcuInstance that may vary in different Variants of the Ecu. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
ethSwitchPort Group Derivation	Boolean	0..1	attr	Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.
sleepMode Supported	Boolean	1	attr	Specifies whether the ECU instance may be put to a "low power mode" <ul style="list-style-type: none"> • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software.
v2xSupported	V2xSupportEnum	0..1	attr	This attribute is used to control the existence of the V2X stack on the given EcuInstance.
wakeUpOver BusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

Table 3.1: EcuInstance

Class	EcuInstanceProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	This element describes additional properties of the EcuInstance that may vary in different Variants of the Ecu.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
diagnostic Address	PositiveInteger	0..1	attr	An ECU specific ID for responses of diagnostic routines.

Table 3.2: EcuInstanceProps

[constr_3008] EcuInstance subelements [The [CommunicationConnector](#) and the [CommunicationController](#) that is referenced by the [CommunicationConnector](#) shall be owned by the same [EcuInstance](#).] ()

Class	ClientIdRange			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	With this element it is possible to restrict the Client Identifier of the transaction handle that is generated by the client RTE for inter-Ecu Client/Server communication to an allowed range of numerical values.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
lowerLimit	Limit	1	attr	This specifies the lower limit of the ClientIdRange. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
upperLimit	Limit	1	attr	This specifies the upper limit of the ClientIdRange. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 3.3: ClientIdRange

[constr_3116] Overlap of ClientIdRanges in the context of the enclosing System [The [ClientIdRange](#) defined for an [EcuInstance](#) shall not overlap with the [ClientIdRange](#) of any other [EcuInstance](#) in the context of the enclosing System.] ()

3.1.2 Communication Controller

[TPS_SYST_01007] Definition of CommunicationController [A [CommunicationController](#) is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium.] ([RS_SYST_00013](#))

[TPS_SYST_01008] Assign CommunicationController to the AUTOSAR Communication Peripheral [In order to illustrate the relationship of an [CommunicationController](#) to the [HwElement](#) with [category](#) [CommunicationController](#) defined in the ECU Resource Description, a mapping between these two classes may be specified using the [CommunicationControllerMapping](#) (see [\[TPS_SYST_01014\]](#)).] ([RS_SYST_00013](#))

Class	<<atpVariation>> CommunicationController (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication controller is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium. Tags: vh.latestBindingTime=postBuild			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractCanCommunicationController, EthernetCommunicationController, FlexrayCommunicationController, LinCommunicationController, UserDefinedCommunicationController			
Attribute	Type	Mult.	Kind	Note
wakeUpByControllerSupported	Boolean	0..1	attr	Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.

Table 3.4: CommunicationController

An [EcuInstance](#) may be connected to the same [PhysicalChannel](#) via two or more [CommunicationControllers](#). In most cases each of these [CommunicationControllers](#) will have a dedicated [CommunicationConnector](#).

There may be rare use cases where an [EcuInstance](#) is connected to a [PhysicalChannel](#) by one [CommunicationController](#) that in turn uses more than one [CommunicationConnector](#).

3.1.3 Communication Connector

[TPS_SYST_01009] Definition of [CommunicationConnector](#) [An [EcuInstance](#) uses [CommunicationConnector](#) elements in order to describe its bus interfaces and to specify the sending/receiving behavior.] ([RS_SYST_00013](#))

The relationship between an [EcuInstance](#), a [CommunicationController](#), and a [PhysicalChannel](#) is expressed by letting a [PhysicalChannel](#) reference a [CommunicationConnector](#) (which in turn is aggregated by [EcuInstance](#)) and which also has the ability to reference a [CommunicationController](#).

Class	CommunicationConnector (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.





Class	CommunicationConnector (abstract)			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	AbstractCanCommunicationConnector , EthernetCommunicationConnector , FlexrayCommunicationConnector , LinCommunicationConnector , UserDefinedCommunicationConnector			
Attribute	Type	Mult.	Kind	Note
commController	CommunicationController	1	ref	Reference to the communication controller. The CommunicationConnector and referenced CommunicationController shall be aggregated by the same ECUInstance. The communicationController can be referenced by several CommunicationConnector elements. This is important for the FlexRay Bus. FlexRay communicates via two physical channels. But only one controller in an ECU is responsible for both channels. Thus, two connectors (for channel A and for channel B) shall reference to the same controller.
createEcuWakeupSource	Boolean	0..1	attr	If this parameter is available and set to true then a channel wakeup source shall be created for the PhysicalChannel referencing this CommunicationConnector.
ecuCommPortInstance	CommConnectorPort	*	aggr	An ECUs reception or send ports. atpVariation: If signals/PDUs/frames are variable, the corresponding ports shall be variable, too. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
pncGatewayType	PncGatewayTypeEnum	0..1	attr	Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".

Table 3.5: CommunicationConnector

Enumeration	PncGatewayTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	Defines the PncGateway roles.
Literal	Description
active	The active PncGateway functionality shall be performed Tags: atp.EnumerationLiteralIndex=0
none	No PncGateway functionality shall be performed Tags: atp.EnumerationLiteralIndex=1
passive	The passive PncGateway functionality shall be performed Tags: atp.EnumerationLiteralIndex=2

Table 3.6: PncGatewayTypeEnum

Note: Use-case for the relation of several [CommunicationConnectors](#) assigned to one [PhysicalChannel](#) in the scope of one [EcuInstance](#): One safety measure for a safety relevant ECU can be to have two transceivers (and two controllers) connected to the same network (Bus). In case a safety violation is detected one transceiver can be disabled and the respective Frames are blocked. The other transceiver stays active and keeps the ECU alive for diagnostics.

3.2 Communication Clustering

3.2.1 Communication Cluster

[TPS_SYST_01010] **Definition of CommunicationCluster** [CommunicationCluster represents a formal way to express that a number of EcuInstances are linked by an arbitrary topology (bus, star, ring, tree). Depending on the communication standard, a CommunicationCluster may either have exactly one or more (redundant) PhysicalChannels.] (RS_SYST_00013)

Note that all ECUs within a CommunicationCluster communicate within the same address range.

Note that the same ECU can participate in more than one CommunicationCluster if it has more than one CommunicationConnector being referenced by PhysicalChannels owned by different CommunicationClusters.

Class	<<atpVariation>> CommunicationCluster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	<p>The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.</p> <p>A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both.</p> <p>A CommunicationCluster aggregates one or more physical channels.</p> <p>Tags:vh.latestBindingTime=postBuild</p>			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	AbstractCanCluster, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Attribute	Type	Mult.	Kind	Note
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
physical Channel	PhysicalChannel	1..*	aggr	<p>This relationship defines which channel element belongs to which cluster. A channel shall be assigned to exactly one cluster, whereas a cluster may have one or more channels.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p>
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

Table 3.7: CommunicationCluster

Some communication clusters need, additional to the general attributes which are valid for all communication clusters, specialized attributes to describe the individual communication cluster properties. The bustype-specific specializations of CommunicationCluster (Figure 3.3) are further detailed in chapter 3.3.

3.2.2 Physical Channel

[TPS_SYST_01011] Definition of **PhysicalChannel** [*PhysicalChannel* represents the communication medium that is used to send and receive information between communicating ECUs. Each *CommunicationCluster* has at least one *PhysicalChannel*.] (*RS_SYST_00013*)

[constr_3373] Limitation on the number of **PhysicalChannels** that are referencing a **CommunicationConnector** [A *CommunicationConnector* shall only be referenced by at most one *PhysicalChannel*.] ()

Class	<i>PhysicalChannel</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	<p>A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each <i>CommunicationCluster</i> has at least one physical channel. Bus systems like CAN and LIN only have exactly one <i>PhysicalChannel</i>. A FlexRay cluster may have more than one <i>PhysicalChannels</i> that may be used in parallel for redundant communication.</p> <p>An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.#</p>			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>AbstractCanPhysicalChannel</i> , <i>EthernetPhysicalChannel</i> , <i>FlexrayPhysicalChannel</i> , <i>LinPhysicalChannel</i> , <i>UserDefinedPhysicalChannel</i>			
Attribute	Type	Mult.	Kind	Note
comm Connector	<i>CommunicationConnector</i>	*	ref	<p>Reference to the <i>ECUInstance</i> via a <i>CommunicationConnector</i> to which the channel is connected.</p> <p>atpVariation: Variable assignment of <i>PhysicalChannels</i> to different <i>CommunicationConnectors</i> is expressed with this variation.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild</p>
frameTriggering	<i>FrameTriggering</i>	*	aggr	<p>One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
iSignal Triggering	<i>ISignalTriggering</i>	*	aggr	<p>One <i>ISignalTriggering</i> is defined for exactly one channel. Channels may have assigned an arbitrary number of <i>ISignaltriggerings</i>.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
managed Physical Channel	<i>PhysicalChannel</i>	*	ref	Reference between a channel with role managing channel and a channel with role managed channel.





Class	PhysicalChannel (abstract)			
pduTriggering	PduTriggering	*	aggr	<p>One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>

Table 3.8: PhysicalChannel

3.3 Specialized Attributes of the Topology Entities

According to their characteristic features, different communication standards like FlexRay, CAN, TTCAN, LIN, J1939 and Ethernet have individual attributes that need to be described additionally to the common topology classes. Figure 3.3 shows the specialization of the [CommunicationCluster](#) into the more specific [FlexrayCluster](#), [CanCluster](#), [TtcanCluster](#), [J1939Cluster](#), [LinCluster](#) and [EthernetCluster](#).

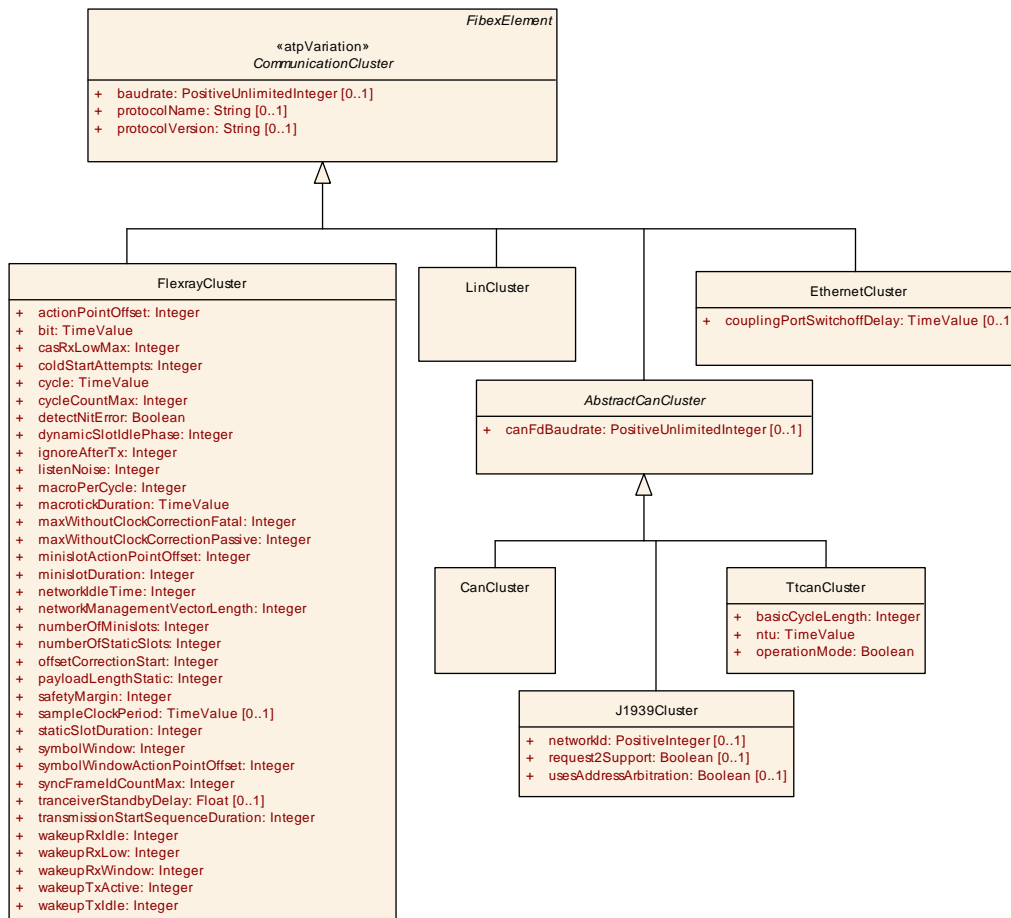


Figure 3.3: Specialized CommunicationCluster attributes (TopologyAttributeRefinement)

3.3.1 CAN

Modeling of the CAN bus is supported in the System Template by the means of four specialized meta-model classes: [CanCluster](#), [CanCommunicationController](#), [CanPhysicalChannel](#), [CanCommunicationConnector](#) (Figure 3.4).

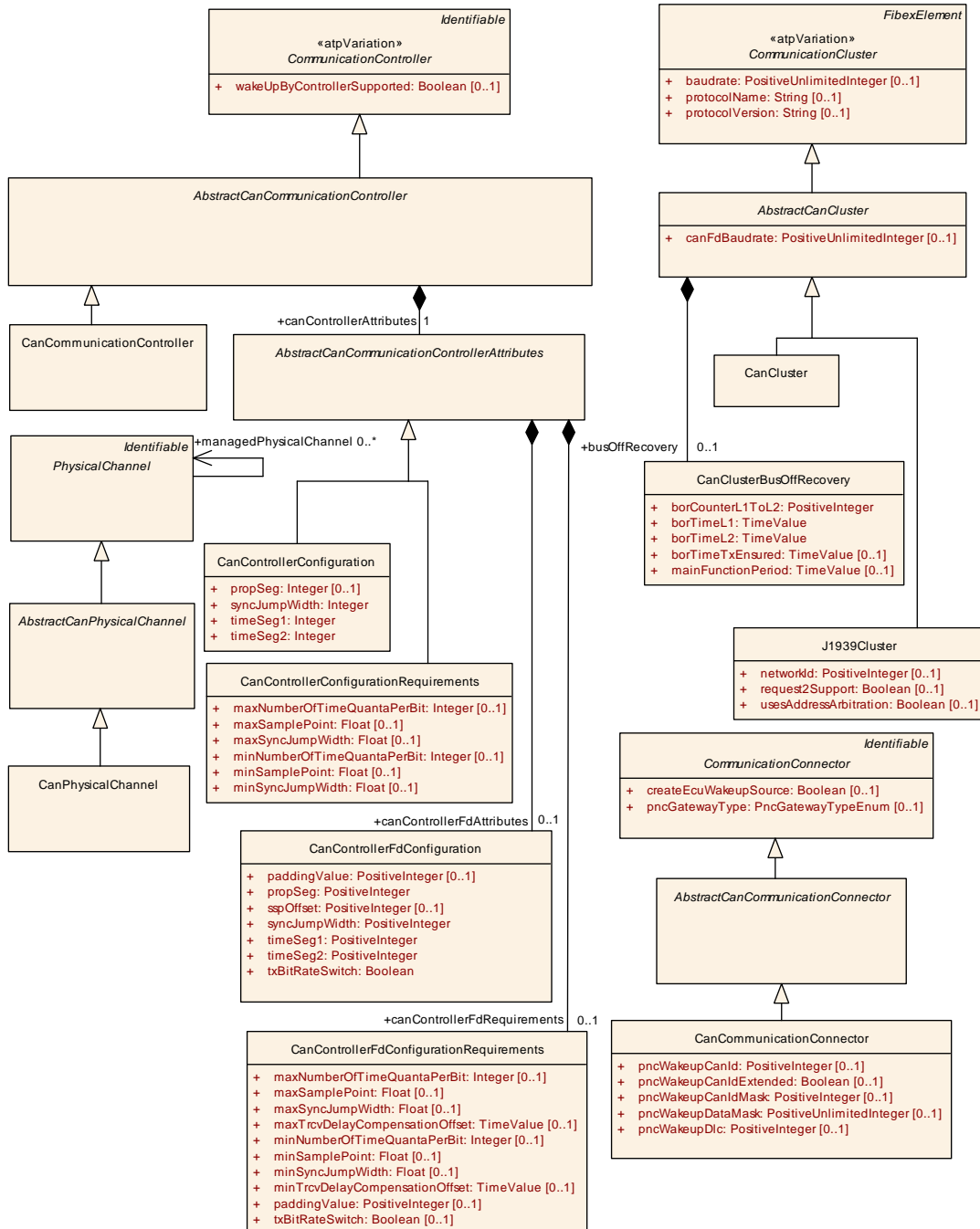


Figure 3.4: CAN bus elements (Fibex4Can_Topology)

3.3.1.1 CAN Cluster

[CanCluster](#) specifies the existence of a CAN cluster in the system's physical topology. It contains additional CAN-specific cluster-wide attributes. The common CAN and TTCAN attributes are collected in the [AbstractCanCluster](#) class.

Class	<<atpVariation>> AbstractCanCluster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN, J1939 and CAN Cluster attributes.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>CommunicationCluster</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	CanCluster , J1939Cluster , TcanCluster			
Attribute	Type	Mult.	Kind	Note
busOffRecovery	CanClusterBusOffRecovery	0..1	aggr	CAN bus off monitoring / recovery at system level.
canFdBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the controller in bits/s.

Table 3.9: AbstractCanCluster

Class	<<atpVariation>> CanCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific cluster attributes. Tags: atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject</i> , AbstractCanCluster , <i>CollectableElement</i> , <i>CommunicationCluster</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.10: CanCluster

Class	CanClusterBusOffRecovery			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element contains the attributes that are used to configure the CAN bus off monitoring / recovery at system level.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
borCounterL1ToL2	PositiveInteger	1	attr	This threshold defines the count of bus-offs until the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).
borTimeL1	TimeValue	1	attr	This attribute defines the duration of the bus-off recovery time in level 1 (short recovery time) in seconds.
borTimeL2	TimeValue	1	attr	This attribute defines the duration of the bus-off recovery time in level 2 (long recovery time) in seconds.
borTimeTxEnsured	TimeValue	0..1	attr	This attribute defines the duration of the bus-off event check in seconds.





Class	CanClusterBusOffRecovery			
mainFunctionPeriod	TimeValue	0..1	attr	This attribute defines the cycle time of the function CanSM_MainFunction in seconds.

Table 3.11: CanClusterBusOffRecovery

3.3.1.2 CAN Communication Controller

[CanCommunicationController](#) is a specialization of the abstract [CommunicationController](#) class. It contains the specific CAN controller attributes needed for configuring the CAN stack in an ECU connected to a certain CAN cluster. The common CAN and TTCAN attributes are collected in the [AbstractCanCommunicationController](#) class. It is possible to specify the CAN Controller configuration parameters as exact values or as requirements that have to be respected by the ECU developer. Therefore the two elements [CanControllerConfiguration](#) and [CanControllerConfigurationRequirements](#) were created.

Class	<<atpVariation>> CanCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication port attributes.			
Base	ARObject , AbstractCanCommunicationController , CommunicationController , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
-	-	-	-	-

Table 3.12: CanCommunicationController

Class	<<atpVariation>> AbstractCanCommunicationController (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN and CAN Controller attributes.			
Base	ARObject , CommunicationController , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	CanCommunicationController , TtcanCommunicationController			
Attribute	Type	Mult.	Kind	Note
canControllerAttributes	AbstractCanCommunicationControllerAttributes	1	aggr	CAN Bit Timing configuration

Table 3.13: AbstractCanCommunicationController

Class	AbstractCanCommunicationControllerAttributes (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	For the configuration of the CanController parameters two different approaches can be used: <ol style="list-style-type: none"> 1. Providing exact values which are taken by the ECU developer (CanControllerConfiguration). 2. Providing ranges of values which are taken as requirements and have to be respected by the ECU developer (CanControllerConfigurationRequirements). 			
Base	ARObject			
Subclasses	CanControllerConfiguration , CanControllerConfigurationRequirements			
Attribute	Type	Mult.	Kind	Note
canControllerFdAttributes	CanControllerFdConfiguration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanFD frame. If this element exists the controller supports CanFD frames and the ECU developer shall take these values for the configuration of the CanFD controller.
canControllerFdRequirements	CanControllerFdConfigurationRequirements	0..1	aggr	Additional CanFD ranges of the bit timing related configuration of a CanFD controller. If this element exists the controller supports CanFD frames and the ECU developer shall take these ranges as requirements for the configuration of the CanFD controller.

Table 3.14: AbstractCanCommunicationControllerAttributes

Class	CanControllerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element is used for the specification of the exact CAN Bit Timing configuration parameter values.			
Base	ARObject, AbstractCanCommunicationControllerAttributes			
Attribute	Type	Mult.	Kind	Note
propSeg	Integer	0..1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	Integer	1	attr	The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
timeSeg1	Integer	1	attr	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
timeSeg2	Integer	1	attr	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2

Table 3.15: CanControllerConfiguration

Class	CanControllerConfigurationRequirements			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element allows the specification of ranges for the CAN Bit Timing configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	ARObject, AbstractCanCommunicationControllerAttributes			
Attribute	Type	Mult.	Kind	Note





Class	CanControllerConfigurationRequirements			
maxNumberOfTimeQuantaPerBit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSamplePoint	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJumpWidth	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minNumberOfTimeQuantaPerBit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJumpWidth	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Table 3.16: CanControllerConfigurationRequirements

[TPS_SYST_01154] CAN Controller support of CAN FD frames [The bit timing configuration of CAN controllers for CAN FD frames is supported by the `CanControllerFdConfiguration` element that is aggregated by `AbstractCanCommunicationControllerAttributes`.] (*RS_SYST_00048*)

[constr_3095] canControllerFdAttributes and canControllerFdRequirements are mutually exclusive [The existence of `canControllerFdAttributes` and `canControllerFdRequirements` is mutually exclusive.] ()

[constr_3518] Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue [The value given for `CanControllerFdConfiguration.paddingValue` and `CanControllerFdConfigurationRequirements.paddingValue` shall be in the range from 0 to 255.] ()

Class	CanControllerFdConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Bit timing related configuration of a CAN controller for payload and CRC of a CAN FD frame.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.
propSeg	PositiveInteger	1	attr	Specifies propagation delay in time quantas.





Class	CanControllerFdConfiguration			
sspOffset	PositiveInteger	0..1	attr	Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
syncJumpWidth	PositiveInteger	1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	1	attr	Specifies phase segment 2 in time quantas.
txBitRateSwitch	Boolean	1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table 3.17: CanControllerFdConfiguration

Class	CanControllerFdConfigurationRequirements			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element allows the specification of ranges for the CanFD bit timing configuration parameters. These ranges are taken as requirements and shall be respected by the ECU developer.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
maxNumberOfTimeQuantaPerBit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSamplePoint	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJumpWidth	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
maxTrcvDelayCompensationOffset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOfTimeQuantaPerBit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.





<i>Class</i>	CanControllerFdConfigurationRequirements			
minSyncJumpWidth	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minTrcvDelayCompensationOffset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.
txBitRateSwitch	Boolean	0..1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table 3.18: CanControllerFdConfigurationRequirements

3.3.1.3 CAN Physical Channel

`CanPhysicalChannel` is a specialization of the abstract `PhysicalChannel` class. It contains the specific CAN `PhysicalChannel` attributes. The common CAN and TTCAN attributes are collected in the `AbstractCanPhysicalChannel` class.

Class	AbstractCanPhysicalChannel (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TTCAN and CAN PhysicalChannel attributes.			
Base	<code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PhysicalChannel</code> , <code>Referrable</code>			
Subclasses	<code>CanPhysicalChannel</code> , <code>TcanPhysicalChannel</code>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.19: AbstractCanPhysicalChannel

Class	CanPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific physical channel attributes.			
Base	<code>ARObject</code> , <code>AbstractCanPhysicalChannel</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PhysicalChannel</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.20: CanPhysicalChannel

[constr_3003] Number of CAN channels [CAN clusters shall aggregate exactly one `PhysicalChannel`.]()

3.3.1.4 CAN Communication Connector

`CanCommunicationConnector` is a specialization of the abstract `CommunicationConnector` class. It contains the specific CAN `CommunicationConnector` attributes. The common CAN and TTCAN attributes are collected in the `AbstractCanCommunicationConnector` class.

Class	AbstractCanCommunicationConnector (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TTCAN and CAN CommunicationConnector attributes.			
Base	<code>ARObject</code> , <code>CommunicationConnector</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Subclasses	<code>CanCommunicationConnector</code> , <code>TcanCommunicationConnector</code>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.21: AbstractCanCommunicationConnector

Class	CanCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication connector attributes.			
Base	ARObject, AbstractCanCommunicationConnector, CommunicationConnector, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
pncWakeupCanId	PositiveInteger	0..1	attr	CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupCanIdExtended	Boolean	0..1	attr	Defines whether pncWakeupCanId and pncWakeupCanIdMask shall be interpreted as extended or standard CAN ID.
pncWakeupCanIdMask	PositiveInteger	0..1	attr	Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDlc	PositiveInteger	0..1	attr	Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.

Table 3.22: CanCommunicationConnector

[TPS_SYST_02165] Derivation of CanNmPnFilterMaskByte [The pncWakeupDataMask should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs.

Note that for one EcuInstance all contributing CanCommunicationConnector.pncWakeupDataMask will be bitwise Ored to obtain the value of CanNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.] (RS_SYST_00042)

Example: For pncWakeupDataMask = 2^{63} and pncVectorOffset = 2, pncIdentifier with number 63 in a NmPdu will be masked (see Figure 3.5).

NmPdu	Byte 0								Byte 1								Byte 2								Byte 3							
Absolute bit position	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
PNC identifiers				N	O	T					U	S	E	D			23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

NmPdu	Byte 4								Byte 5								Byte 6								Byte 7							
Absolute bit position	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
PNC identifiers	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56

Figure 3.5: Example of masked pncIdentifiers in a NmPdu

3.3.2 TTCAN

Modeling of TTCAN clusters is supported in the System Template by the means of four specialized meta-model classes: TtcanCluster, TtcanCommunicationController, TtcanCommunicationConnector, TtcanPhysicalChannel (figure 3.6).

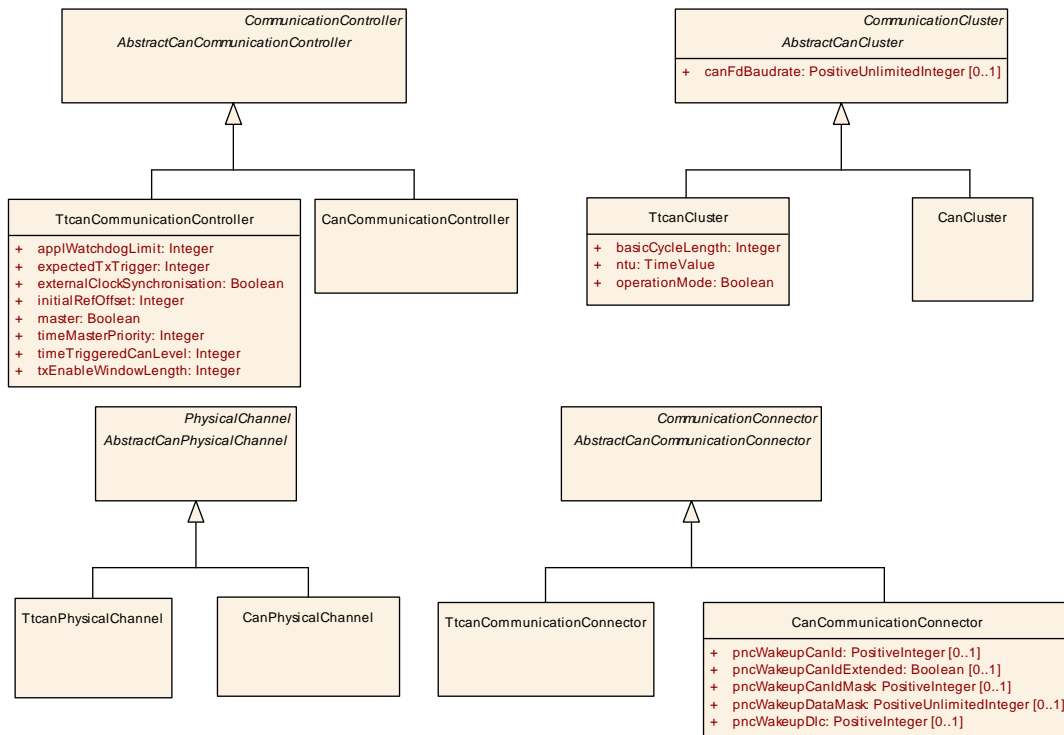


Figure 3.6: TTCAN bus elements (Fibex4Ttcan_Topology)

3.3.2.1 TTCAN Cluster

`TtcanCluster` specifies the existence of a TTCAN cluster in the system’s physical topology. Additionally to the common CAN and TTCAN attributes it contains TTCAN-specific cluster-wide attributes.

Class	<<atpVariation>> TtcanCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific cluster attributes. Tags: atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject</i> , <i>AbstractCanCluster</i> , <i>CollectableElement</i> , <i>CommunicationCluster</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
basicCycleLength	Integer	1	attr	Length of a basic-cycle. Unit: NTUs
ntu	TimeValue	1	attr	Unit measuring all times and providing a constant of the whole network. For level 1, this is always the CAN bit time. Unit: seconds.
operationMode	Boolean	1	attr	Possible operation modes True: Time-Triggered False: Event-Synchronised-Time-Triggered

Table 3.23: TtcanCluster

3.3.2.2 TTCAN Communication Controller

TtcanCommunicationController is a specialization of the *AbstractCanCommunicationController* class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN Controller attributes.

Class	<<atpVariation>> TtcanCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific communication port attributes.			
Base	<i>ARObject</i> , <i>AbstractCanCommunicationController</i> , <i>CommunicationController</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
applWatchdogLimit	Integer	1	attr	The Appl_Watchdog_Limit shall be an 8-bit value specifying the period for the application watchdog in Appl_Watchdog_Limit times 256 NTUs.
expectedTxTrigger	Integer	1	attr	The Expected_Tx_Trigger shall be an eight (8) bit value which limits the number of messages the FSE may try to transmit in one matrix cycle.
externalClockSynchronisation	Boolean	1	attr	One bit shall be used to configure whether or not external clock synchronisation will be allowed during runtime (only Level 2).
initialRefOffset	Integer	1	attr	The Initial_Ref_Offset shall be an eight (8) bit value for the initialisation of Ref_Trigger_Offset.
master	Boolean	1	attr	One bit shall be used to distinguish between (potential) time masters and time slaves. This can be derived from the frame-triggering's triggers.





Class	<<atpVariation>> TtcanCommunicationController			
timeMasterPriority	Integer	1	attr	The time master priority shall contain a three bit value for the priority of the current time master (the last three bits of the identifier of the reference message). This can be derived from the frame-triggering's triggers.
timeTriggeredCanLevel	Integer	1	attr	One bit shall be used to distinguish between Level 1 and Level 2.
txEnableWindowLength	Integer	1	attr	The length of the Tx_Enable window shall be a four (4) bit value specifying the length of the time period (1-16 nominal CAN bit times) in which a transmission may be started.

Table 3.24: TtcanCommunicationController

3.3.2.3 TTCAN Physical Channel

`TtcanPhysicalChannel` is a specialization of the `AbstractCanPhysicalChannel` class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN Physical Channel attributes.

Class	TtcanPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific physical channel attributes.			
Base	<i>ARObject</i> , <i>AbstractCanPhysicalChannel</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PhysicalChannel</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.25: TtcanPhysicalChannel

3.3.2.4 TTCAN Communication Connector

`TtcanCommunicationConnector` is a specialization of the `AbstractCanCommunicationConnector` class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN `CommunicationConnector` attributes.

Class	TtcanCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific communication connector attributes.			
Base	<i>ARObject</i> , <i>AbstractCanCommunicationConnector</i> , <i>CommunicationConnector</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–





Class	TtcanCommunicationConnector			
-	-	-	-	-

Table 3.26: TtcanCommunicationConnector

3.3.3 SAE J1939

Modeling of J1939 Communication Clusters is supported in the System Template with the `J1939Cluster` element that is derived from `AbstractCanCluster` (see figure 3.4).

Class	<<atpVariation>> J1939Cluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	J1939 specific cluster attributes. Tags: atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject, AbstractCanCluster, CollectableElement, CommunicationCluster, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
networkId	PositiveInteger	0..1	attr	This represents the network ID for the J1939 cluster.
request2Support	Boolean	0..1	attr	Enables support for the Request2 PGN (RQST2).
usesAddress Arbitration	Boolean	0..1	attr	Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes. True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not care for contending address claims.

Table 3.27: J1939Cluster

To describe the communication on a `J1939Cluster` `CanFrameTriggerings` are used that are aggregated by a `CanPhysicalChannel`.

[constr_3050] `J1939Cluster` uses exactly one `CanPhysicalChannel` [A `J1939Cluster` shall aggregate exactly one `CanPhysicalChannel`.]()

[constr_1463] Applicable values for `J1939Cluster.networkId` [The values of the attribute `J1939Cluster.networkId` shall always be within the interval 1..4.]()

Please note that AUTOSAR supports only the four mentioned bus types. Still, an implementation could e.g. support J1708 [14] by means of a complex driver and would then need to assign the corresponding bus type.

3.3.4 FlexRay

Modeling of FlexRay clusters is supported in the System Template by the means of four specialized meta-model classes: `FlexrayCluster`, `FlexrayCommunicationConnector`, `FlexrayPhysicalChannel`, `FlexrayCommunicationController` (Figure 3.7).

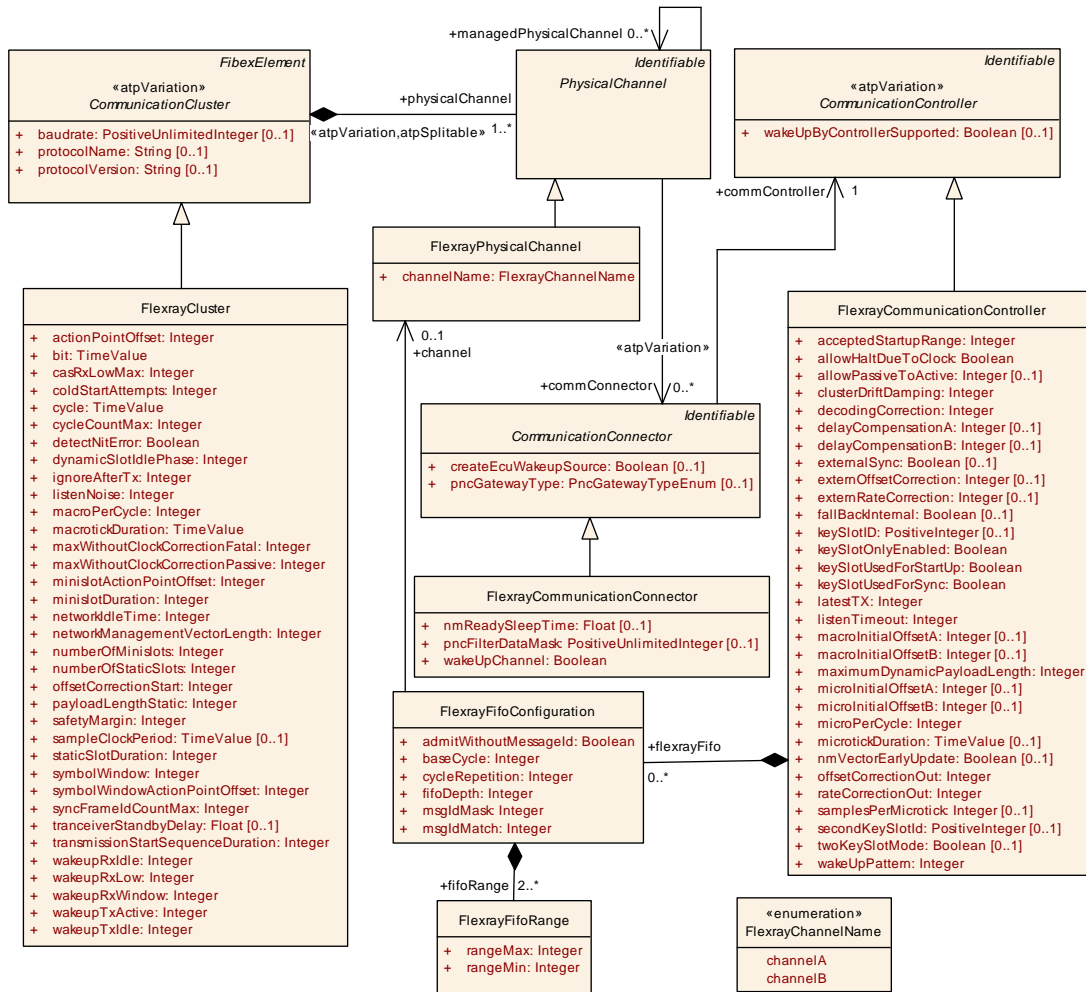


Figure 3.7: FlexRay cluster elements (Fibex4FlexRay_Topology)

3.3.4.1 FlexRay Cluster

`FlexrayCluster` specifies the existence of a FlexRay cluster in the system's physical topology. It contains additional FlexRay-specific cluster-wide attributes.

Class	<<atpVariation>> FlexrayCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the physicalCluster Tags: atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject, CollectableElement, CommunicationCluster, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
actionPointOffset	Integer	1	attr	The offset of the action point in networks
bit	TimeValue	1	attr	Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPer Bit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLowMax	Integer	1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStartAttempts	Integer	1	attr	The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization
cycle	TimeValue	1	attr	Length of the cycle. Unit: seconds
cycleCountMax	Integer	1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
detectNitError	Boolean	1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSlotIdlePhase	Integer	1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfterTx	Integer	1	attr	Duration for which the bitstrobing is paused after transmission [gdBit].
listenNoise	Integer	1	attr	Upper limit for the start up and wake up listen timeout in the presence of noise. Expressed as a multiple of the cluster constant pdListenTimeout. Unit microticks
macroPerCycle	Integer	1	attr	The number of macroticks in a communication cycle
macrotickDuration	TimeValue	1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithoutClockCorrectionFatal	Integer	1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state.
maxWithoutClockCorrectionPassive	Integer	1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state.
minislotActionPointOffset	Integer	1	attr	The Offset of the action point within a minislot. Unit: macroticks
minislotDuration	Integer	1	attr	The duration of a minislot (dynamic segment). Unit: macroticks.
networkIdleTime	Integer	1	attr	The duration of the network idle time in macroticks
networkManagementVectorLength	Integer	1	attr	Length of the Network Management vector in a cluster [bytes]
numberOfMinislots	Integer	1	attr	Number of Minislots in the dynamic segment.
numberOfStaticSlots	Integer	1	attr	The number of static slots in the static segment.





Class	<<atpVariation>> FlexrayCluster			
offsetCorrectionStart	Integer	1	attr	Start of the offset correction phase within the Network Idle Time (NIT), expressed as the number of macroticks from the start of cycle. Unit: macroticks
payloadLengthStatic	Integer	1	attr	Globally configured payload length of a static frame. Unit: 16-bit WORDS.
safetyMargin	Integer	1	attr	Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has been resynchronized.
sampleClockPeriod	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlotDuration	Integer	1	attr	The duration of a slot in the static segment. Unit: macroticks
symbolWindow	Integer	1	attr	The duration of the symbol window. Unit: macroticks
symbolWindowActionPointOffset	Integer	1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameIdCountMax	Integer	1	attr	Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.
transceiverStandbyDelay	Float	0..1	attr	The duration of timer t_TrvcStdbDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value. Not specifying a value or a value of 0 shall imply that the timer is not used.
transmissionStartSequenceDuration	Integer	1	attr	Number of bits in the Transmission Start Sequence [gd Bits].
wakeupRxIdle	Integer	1	attr	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit: bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.
wakeupRxLow	Integer	1	attr	Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit: bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow.
wakeupRxWindow	Integer	1	attr	The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow.
wakeupTxActive	Integer	1	attr	Number of bits used by the node to transmit the LOW phase of a wakeup symbol and the HIGH and LOW phases of a WUDOP. Unit: bitDuration
wakeupTxIdle	Integer	1	attr	Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gdBit

Table 3.28: FlexrayCluster

3.3.4.2 FlexRay Communication Controller

`FlexrayCommunicationController` is a specialization of the `CommunicationController` class. It contains the specific FlexRay controller attributes needed for configuring the FlexRay stack in an ECU connected to a certain FlexRay cluster.

Class	<<atpVariation>> FlexrayCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay bus specific communication port attributes.			
Base	<code>ARObject</code> , <code>CommunicationController</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
acceptedStartupRange	Integer	1	attr	Expanded range of measured clock deviation allowed for startup frames during integration. Unit: microtick
allowHaltDueToClock	Boolean	1	attr	Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the Communication Controller is allowed to transition to POC:halt. If set to false, the Communication Controller will not transition to the POC:halt state but will enter or remain in the normal POC (passive State).
allowPassiveToActive	Integer	0..1	attr	Number of consecutive even/odd cycle pairs that shall have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to 0, the Communication Controller is not allowed to transition from POC:norm
clusterDriftDamping	Integer	1	attr	The cluster drift damping factor used in clock synchronization rate correction in microticks
decodingCorrection	Integer	1	attr	Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)
delayCompensationA	Integer	0..1	attr	Value used to compensate for reception delays on channel A Unit: Microticks. This optional parameter shall only be filled out if channel A is used.
delayCompensationB	Integer	0..1	attr	Value used to compensate for reception delays on channel B. Unit: Microticks. This optional parameter shall only be filled out if channel B is used.
externalSync	Boolean	0..1	attr	Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TTE Time Triggered External Sync cluster) or locally synchronized.
externOffsetCorrection	Integer	0..1	attr	Fixed amount added or subtracted to the calculated offset correction term to facilitate external offset correction, expressed in node-local microticks.
externRateCorrection	Integer	0..1	attr	Fixed amount added or subtracted to the calculated rate correction term to facilitate external rate correction, expressed in node-local microticks.
fallBackInternal	Boolean	0..1	attr	Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).
flexrayFifo	FlexrayFifo Configuration	*	aggr	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO.





Class	<<atpVariation>> FlexrayCommunicationController			
keySlotID	PositiveInteger	0..1	attr	ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedForStartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory.
keySlotOnlyEnabled	Boolean	1	attr	Flag indicating whether or not the node shall enter key slot only mode following startup.
keySlotUsedForStartUp	Boolean	1	attr	Flag indicating whether the Key Slot is used to transmit a startup frame.
keySlotUsedForSync	Boolean	1	attr	Flag indicating whether the Key Slot is used to transmit a sync frame.
latestTX	Integer	1	attr	The number of the last minislot in which a transmission can start in the dynamic segment for the respective node
listenTimeout	Integer	1	attr	Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks
macroInitialOffsetA	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel A is used.
macroInitialOffsetB	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel B is used.
maximumDynamicPayloadLength	Integer	1	attr	Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.
microInitialOffsetA	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel A is used.
microInitialOffsetB	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel B is used.
microPerCycle	Integer	1	attr	The nominal number of microticks in a communication cycle
microtickDuration	TimeValue	0..1	attr	Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds
nmVectorEarlyUpdate	Boolean	0..1	attr	Flag indicating when the update of the Network Management Vector in the CHI shall take place. If set to false, the update shall take place after the NIT. If set to true, the update shall take place after the end of the static segment.
offsetCorrectionOut	Integer	1	attr	Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)





Class		<<atpVariation>> FlexrayCommunicationController		
rateCorrection Out	Integer	1	attr	Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrection Out) Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift.
samplesPer Microtick	Integer	0..1	attr	Number of samples per microtick
secondKeySlot Id	PositiveInteger	0..1	attr	ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.
twoKeySlot Mode	Boolean	0..1	attr	Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.
wakeUpPattern	Integer	1	attr	Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster

Table 3.29: FlexrayCommunicationController

Class	FlexrayFifoConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
admitWithoutMessageId	Boolean	1	attr	Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.
baseCycle	Integer	1	attr	FIFO cycle counter acceptance criteria.
channel	FlexrayPhysicalChannel	0..1	ref	Fifo channel admittance criteria.
cycleRepetition	Integer	1	attr	FIFO cycle counter acceptance criteria.
fifoDepth	Integer	1	attr	FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.
fifoRange	FlexrayFifoRange	2..*	aggr	FIFO Frame Id range acceptance criteria.
msgIdMask	Integer	1	attr	FIFO message identifier acceptance criteria (Mask filter).
msgIdMatch	Integer	1	attr	FIFO message identifier acceptance criteria (Match filter).

Table 3.30: FlexrayFifoConfiguration

Class	FlexrayFifoRange			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FIFO Frame Id range acceptance criteria.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
rangeMax	Integer	1	attr	Max Range.
rangeMin	Integer	1	attr	Min Range.

Table 3.31: FlexrayFifoRange

3.3.4.3 FlexRay Communication Connector

`FlexrayCommunicationConnector` adds the FlexRay specific attributes to the `CommunicationConnector`.

Class	FlexrayCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	<i>ARObject</i> , <i>CommunicationConnector</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
nmReadySleepTime	Float	0..1	attr	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.
pncFilterDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup.
wakeUpChannel	Boolean	1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table 3.32: FlexrayCommunicationConnector

[constr_3508] Value of nmReadySleepTime [The nmReadySleepTime value shall be a multiple of $\text{cycle} * \text{nmRepetitionCycle}$.]()

[TPS_SYST_02167] Derivation of FrNmPnFilterMaskByte [The FrNmPnFilterMaskByte should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs.

Note that for one EcuInstance all contributing FlexrayCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.](RS_SYST_00042)

The masking of pncIdentifiers in a NmPdu based on the pncFilterDataMask is done in the same way as for the CanCommunicationConnector (see Example in Figure 3.5).

3.3.4.4 FlexRay Physical Channel

`FlexrayPhysicalChannel` adds the FlexRay specific attributes to the `PhysicalChannel`.

Class	FlexrayPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the physicalChannel			
Base	ARObject, Identifiable , MultilanguageReferrable , PhysicalChannel , Referrable			
Attribute	Type	Mult.	Kind	Note
channelName	FlexrayChannelName	1	attr	Name of the channel (Channel A or Channel B).

Table 3.33: FlexrayPhysicalChannel

Enumeration	FlexrayChannelName			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	Name of the channel.			
Literal	Description			
channelA	Channel A Tags: atp.EnumerationLiteralIndex=0			
channelB	Channel B Tags: atp.EnumerationLiteralIndex=1			

Table 3.34: FlexrayChannelName

[constr_3018] Number of FlexRay channels [A [FlexrayCluster](#) shall use either one [FlexrayPhysicalChannel](#) with [channelName](#) set to either [channelA](#) or [channelB](#) or else two [FlexrayPhysicalChannels](#) with one [channelName](#) [channelA](#) and one [channelName](#) [channelB](#).]()

3.3.5 LIN

A `LinCluster` consists of exactly one master node connected to several slave nodes. The master is responsible for providing the frame headers on the bus according to a predefined schedule, whereas the slaves send or receive the actual frame information ([10]).

[TPS_SYST_01012] Different Properties of `LinMaster` and `LinSlave` [In the System Template the different properties of master and slave nodes are handled by deriving the LIN-specific subclasses `LinMaster` and `LinSlave` as specializations of `LinCommunicationController`.] (*RS_SYST_00022*)

AUTOSAR supports the stand-alone definition of both LIN masters and LIN slaves.

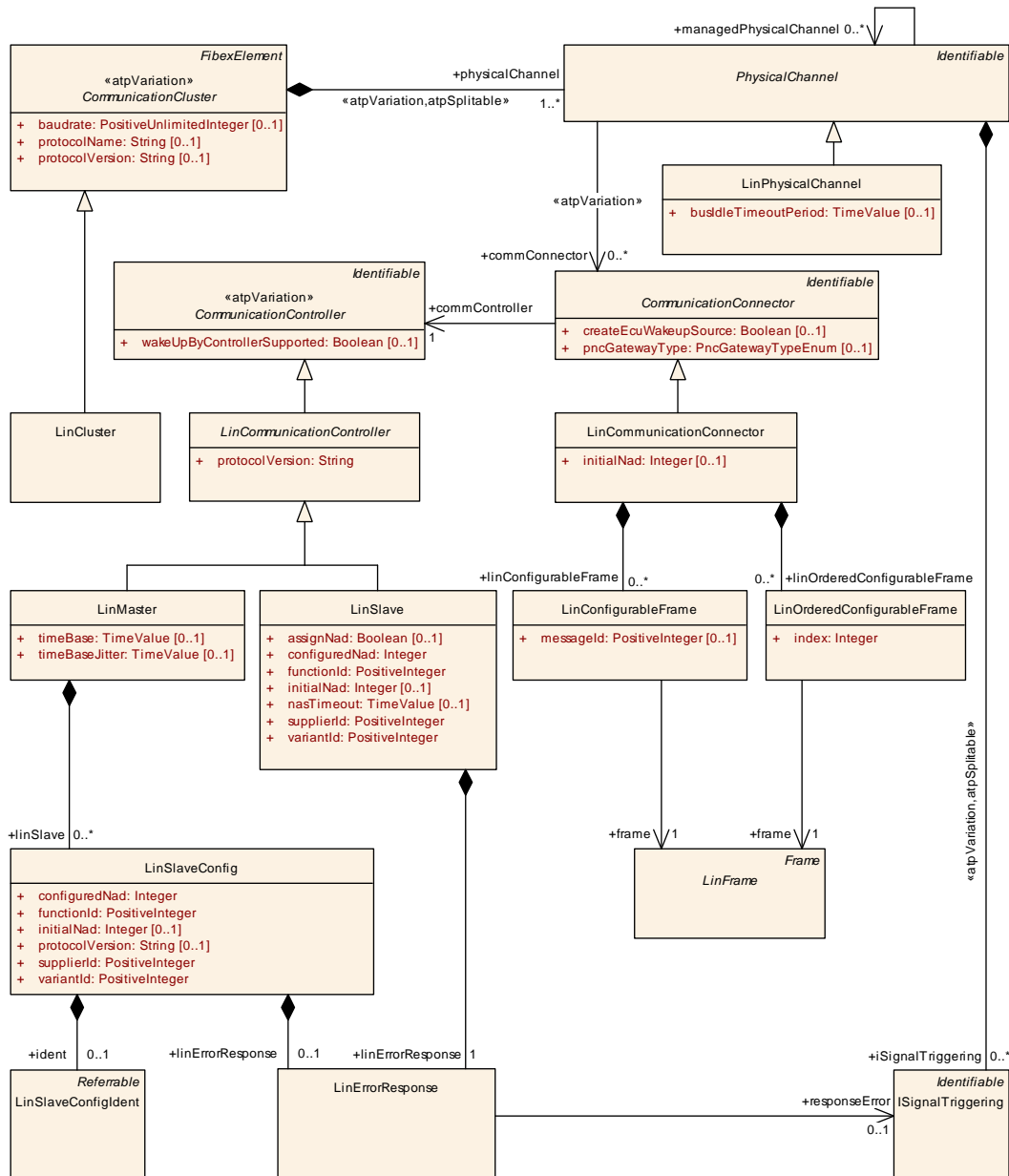


Figure 3.8: Specialized `LinCommunicationController` attributes (Fibex4Lin_Topology)

Note that the AUTOSAR BSW only supports LIN masters. LIN slaves are seen as non AUTOSAR ECUs. They can be described in the System Template in order to configure the LIN Interface for the master correctly, but AUTOSAR does not support the development of LIN slaves as of AUTOSAR release 4.0 ([15], [16]).

3.3.5.1 LIN Cluster

[LinCluster](#) specifies the existence of a LIN cluster in the system's physical topology.

Class	<<atpVariation>> LinCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes Tags: atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>CommunicationCluster</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.35: LinCluster

3.3.5.2 LIN Communication Controller

[LinCommunicationController](#) is a specialization of the [CommunicationController](#) class. It is an abstract class, to be further specialized by [LinMaster](#) and [LinSlave](#).

Class	<<atpVariation>> LinCommunicationController (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication controller attributes.			
Base	<i>ARObject</i> , <i>CommunicationController</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	LinMaster , LinSlave			
Attribute	Type	Mult.	Kind	Note
protocolVersion	String	1	attr	Version specifier for a communication protocol.

Table 3.36: LinCommunicationController

[TPS_SYST_02257] Standardized values of [LinCommunicationController.protocolVersion](#) and [LinSlaveConfig.protocolVersion](#) [The following values of attributes [LinCommunicationController.protocolVersion](#) and [LinSlaveConfig.protocolVersion](#) are standardized by AUTOSAR:

- LIN13
- LIN20
- LIN21

- LIN22
- ISO17987

|(RS_SYST_00022)

3.3.5.3 LIN Master

`LinMaster` describes the existence of a LIN master task in a LIN topology node. As such it contains the attributes specific to a LIN master task.

Class	<<atpVariation>> LinMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Describing the properties of the referring ecu as a LIN master.			
Base	<i>ARObject</i> , <i>CommunicationController</i> , <i>Identifiable</i> , <i>LinCommunicationController</i> , <i>Multilanguage</i> <i>Referrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
linSlave	LinSlaveConfig	*	aggr	LinSlaves that are handled by the LinMaster.
timeBase	TimeValue	0..1	attr	Time base is mandatory for the master. It is not used for slaves. LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time." The time base shall be specified AUTOSAR conform in seconds.
timeBaseJitter	TimeValue	0..1	attr	The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves. LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)." The jitter shall be specified AUTOSAR conform in seconds.

Table 3.37: LinMaster

Class	LinSlaveConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Node attributes of LIN slaves that are handled by the LinMaster. In the System Description LIN slaves may be described in the context of the Lin Master. In an ECU Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the Lin Master. The values of attributes of LinSlaveConfig and the corresponding LinSlave shall be identical (if both are defined in a System Description).			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note





Class	LinSlaveConfig			
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID.
ident	LinSlaveConfigIdent	0..1	aggr	This adds the ability to become referable to LinSlave Config.
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linErrorResponse	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames.
protocolVersion	String	0..1	attr	Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different. Tags: atp.Status=shallBecomeMandatory
supplierId	PositiveInteger	1	attr	LIN Supplier ID.
variantId	PositiveInteger	1	attr	Specifies the Variant ID.

Table 3.38: LinSlaveConfig

Class	LinSlaveConfigIdent			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex4Lin::LinTopology			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referable Lin SlaveConfig.			
Base	<i>ARObject</i> , <i>Referable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.39: LinSlaveConfigIdent

[constr_3219] The mutual existence of LinSlaves in the LinMaster EcuExtract [[LinSlaves](#) shall not be part of the EcuExtract of the corresponding [LinMaster](#).]()

[constr_1655] The mutual existence of LinMasters in the LinSlave EcuExtract [A [LinMaster](#) shall not be part of the EcuExtract of a corresponding [LinSlave](#).]()

[TPS_SYST_02101] Usage of LinSlaveConfig in Ecu Extract [In order to configure [LinMaster](#) in a [System](#) with `category` ECU_EXTRACT the [LinSlaveConfig](#) aggregated by the [LinMaster](#) shall be used.]([RS_SYST_00022](#))

Please note that, in concordance with [\[TPS_SYST_02101\]](#), even if the [LinSlave](#) can be modeled independently of the [LinMaster](#) it still makes sense that the the configuration of the [LinMaster](#) **positively contains the aggregation of the LinSlave-Config** in the role `linSlave`.

In other words, the configuration of a [LinMaster](#) is **not affected** by the question of whether or not the [LinSlave](#) is explicitly modeled.

This statement is valid for both the existence of the [LinMaster](#) in a [System](#) of `category` SYSTEM_DESCRIPTION or in a [System](#) of `category` ECU_EXTRACT.

The actual correspondence between a Lin slave described by means of the `LinSlaveConfig` and the actual model of the `LinSlave` shall be determined by identifying pairs of `LinSlaveConfig` and `LinSlave` with an identical set of the attributes that are equally named in both meta-classes. This rule does not apply for the `shortName`.

Another relevant condition for finding pairs of corresponding `LinSlaveConfig` and `LinSlave` is obviously that the `LinMaster` that aggregates the `LinSlaveConfig` shall be connected to the same `LinCluster` to which the corresponding `LinSlave` is connected.

Of course, this condition can only be checked in the context of a `System` of category `SYSTEM_DESCRIPTION` or perhaps `SYSTEM_EXTRACT`.

3.3.5.4 LIN Slave

AUTOSAR supports the definition of a stand-alone LIN slave¹. In other words, it is possible to define an ECU Extract that contains the modeling of a LIN slave independently of the modeling of the LIN master.

That said, the ability to define properties of the LIN slave in the context of the LIN master in the form of `LinMaster.linSlave` still exists and can be used where applicable.

[TPS_SYST_05018] Semantics of meta-class `LinSlave` [Meta-class `LinSlave` describes the existence of a LIN slave task in a LIN topology node. It describes the attributes of a single LIN slave node.] (*RS_SYST_00022*)

[TPS_SYST_05019] Semantics of `LinErrorResponse.responseError` [Each Lin slave has the ability to set an error bit in the response part of one specific `LinUnconditionalFrame` in the event of errors occurring on frame level. The error bit is modeled by means of a reference to an `ISignalTriggering` in the role `LinErrorResponse.responseError`.] (*RS_SYST_00022*)

Please note that because the response error bit applies for frame errors the responsibility for setting the response error bit lies exclusively at the `LinIf`.

In the event of such an error, the `LinIf` on the Lin slave Ecu calls `Com_SendSignal()` to set the value of the response error bit if applicable² and thus the system model needs to foresee the existence of an `ISignalTriggering` for this purpose.

Aside: on the Lin master, typically a piece of application software picks up the received error bit and uses it to e.g. increment a counter for debouncing purposes. If the counter

¹In former versions of this specification document the properties of a LIN slave could only be defined in the context of the corresponding LIN master

²In principle, the `LinIf` on the Lin slave Ecu could directly patch the value of the response error bit into a Tx Pdu before sending but in this case the `LinIf` would have to pick the correct Pdu and patch it accordingly. In other words, the `LinIf` would replicate a certain amount of `Com` functionality. the usage of an `ISignalTriggering` significantly simplifies the implementation of the `LinIf`.

exceeds a certain value in e.g. a given time interval the Lin master has to assume that a serious problem exists in the communication with this specific slave and react accordingly.

In terms of modeling, this means that if formally modeled application software on the Lin master exists that processes the error response bit (the receiving side of the error response bit) then a [DataMapping](#) to a [SystemSignal](#) that carries the response error bit needs to be defined.

It is important to understand that application software on a Lin slave positively has no business of setting the value of the response error bit. In other words: on the sending side (i.e. the Lin slave), the application software is not affected and therefore there shall not be a [DataMapping](#) on the sending side. This relation motivates the existence of [\[constr_1656\]](#).

[constr_1656] No application-level write access to [LinErrorResponse.responseError](#) on Lin slave [The [SystemSignal](#) referenced in the role [systemSignal](#) by the [ISignal](#) referenced by the [ISignalTriggering](#) that in turn is referenced in the role [LinErrorResponse.responseError](#) shall not be referenced by a [DataMapping](#) that allows for writing to the [SystemSignal](#).]()

Class	<<atpVariation>> LinSlave			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Describing the properties of the referring ecu as a LIN slave.			
Base	ARObject , CommunicationController , Identifiable , LinCommunicationController , Multilanguage Referrable , Referrable			
Attribute	Type	Mult.	Kind	Note
assignNad	Boolean	0..1	attr	This attribute has the ability to control whether the node configuration command 'Assign NAD' is supported.
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID
initialNad	Integer	0..1	attr	This attribute represents the initial NAD.
linError Response	LinErrorResponse	1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames.
nasTimeout	TimeValue	0..1	attr	Value of the N_AS timeout. Unit: seconds.
supplierId	PositiveInteger	1	attr	LIN Supplier ID
variantId	PositiveInteger	1	attr	Specifies the Variant ID

Table 3.40: LinSlave

Class	LinErrorResponse			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Each slave node shall publish a one bit signal, named response_error, to the master node in one of its transmitted unconditional frames. The response_error signal shall be set whenever a frame (except for event triggered frame responses) that is transmitted or received by the slave node contains an error in the frame response. The response_error signal shall be cleared when the unconditional frame containing the response_error signal is successfully transmitted.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
responseError	ISignalTriggering	0..1	ref	This ISignal shall be taken to transport the responseError bit.

Table 3.41: LinErrorResponse

3.3.5.5 LIN Communication Connector

[LinCommunicationConnector](#) is a specialization of the [CommunicationConnector](#) class. The [LinCommunicationConnector](#) element contains lists of frames processed by the slave node.

[constr_3029] Assign-Frame command usage [For the LIN 2.0 Assign-Frame command the [LinConfigurableFrame](#) list shall be used. For the LIN 2.1 Assign-Frame-PID-Range command the [LinOrderedConfigurableFrame](#) list shall be used.]()

[constr_5030] Uniqueness of [LinOrderedConfigurableFrame.index](#) [[LinOrderedConfigurableFrame.index](#) shall always be set and be unique in the context of the aggregating [LinCommunicationConnector](#).]()

Class	LinCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication connector attributes.			
Base	ARObject, CommunicationConnector , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurableFrame	LinConfigurableFrame	*	aggr	LinConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.0 Assign-Frame command.
linOrderedConfigurableFrame	LinOrderedConfigurableFrame	*	aggr	LinOrderedConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.1 Assign-Frame-PID-Range command.

Table 3.42: LinCommunicationConnector

Class	LinConfigurableFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Assignment of messageIds to Frames. This element shall be used for the LIN 2.0 Assign-Frame command.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.
messageId	PositiveInteger	0..1	attr	MessageId for the referenced frame

Table 3.43: LinConfigurableFrame

Class	LinOrderedConfigurableFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	With the assignment of the index to a frame a mapping of Pids to Frames is possible. This element shall be used for the LIN 2.1 Assign-Frame-PID-Range command.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.
index	Integer	1	attr	This attribute is used to order the elements and allows an assignment of Pids to ConfigurableFrames that are defined in the slave.

Table 3.44: LinOrderedConfigurableFrame

3.3.5.6 LIN Physical Channel

[LinPhysicalChannel](#) is a specialization of the [PhysicalChannel](#) class. It contains additional Lin-specific [PhysicalChannel](#) attributes.

Class	LinPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes to the physicalChannel			
Base	ARObject, Identifiable , MultilanguageReferrable , PhysicalChannel , Referrable			
Attribute	Type	Mult.	Kind	Note
busIdleTimeoutPeriod	TimeValue	0..1	attr	This attribute shall be used to set an idle timeout period for the enclosing LinPhysicalChannel.
scheduleTable	LinScheduleTable	*	aggr	Schedule tables organize the timings of the frames for LIN. atpVariation: If the transmitted frames are variable, the corresponding ScheduleTables shall be variable, too. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 3.45: LinPhysicalChannel

[constr_3015] Number of LIN channels [LIN clusters shall aggregate exactly one [LinPhysicalChannel](#).]()

3.3.6 Ethernet

The `EthernetCluster` represents an Ethernet network which may consist of several ECUs connected.

An essential aspect of modern Ethernet is the possibility to introduce Ethernet switches in order to partition the `EthernetCluster` into segments which are used for point-to-point communication between the respective partners. It is possible to define the behavior of such Ethernet switches, this is described in chapter 3.3.6.6.

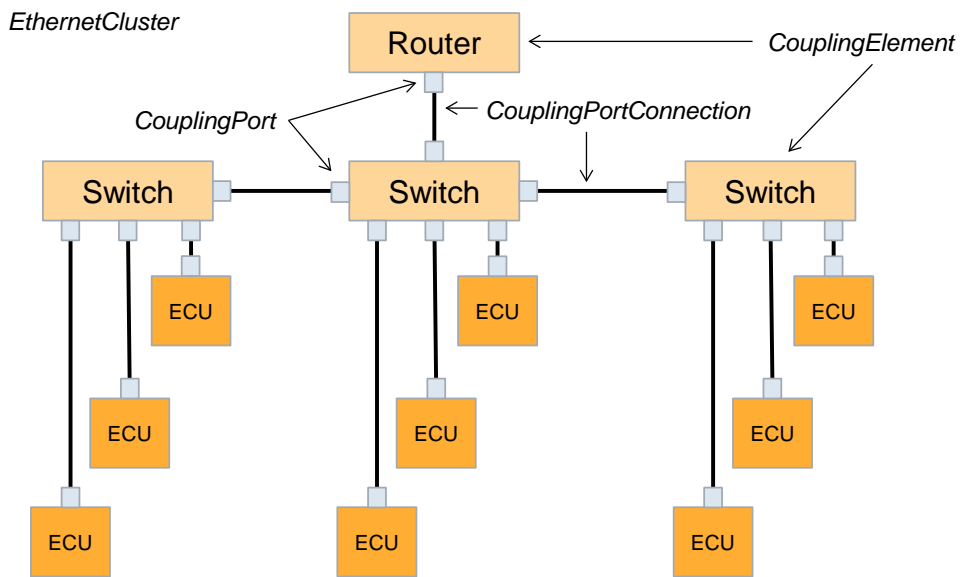


Figure 3.9: Example of an EthernetCluster

Figure 3.9 illustrates an example of an `EthernetCluster`. In this figure the focus is on the *Link Layer* and represents the wiring of ECUs, their communication connectors, switches, hubs, routers, and how these elements are connected electrically.

To describe the Ethernet at the data link- and physical layer the following System Template meta-model classes are used: `EthernetCluster`, `EthernetCommunicationController`, `EthernetCommunicationConnector`, `EthernetPhysicalChannel`, `CouplingElement`, `CouplingPort` and `CouplingPortConnection` (see Figure 3.10).

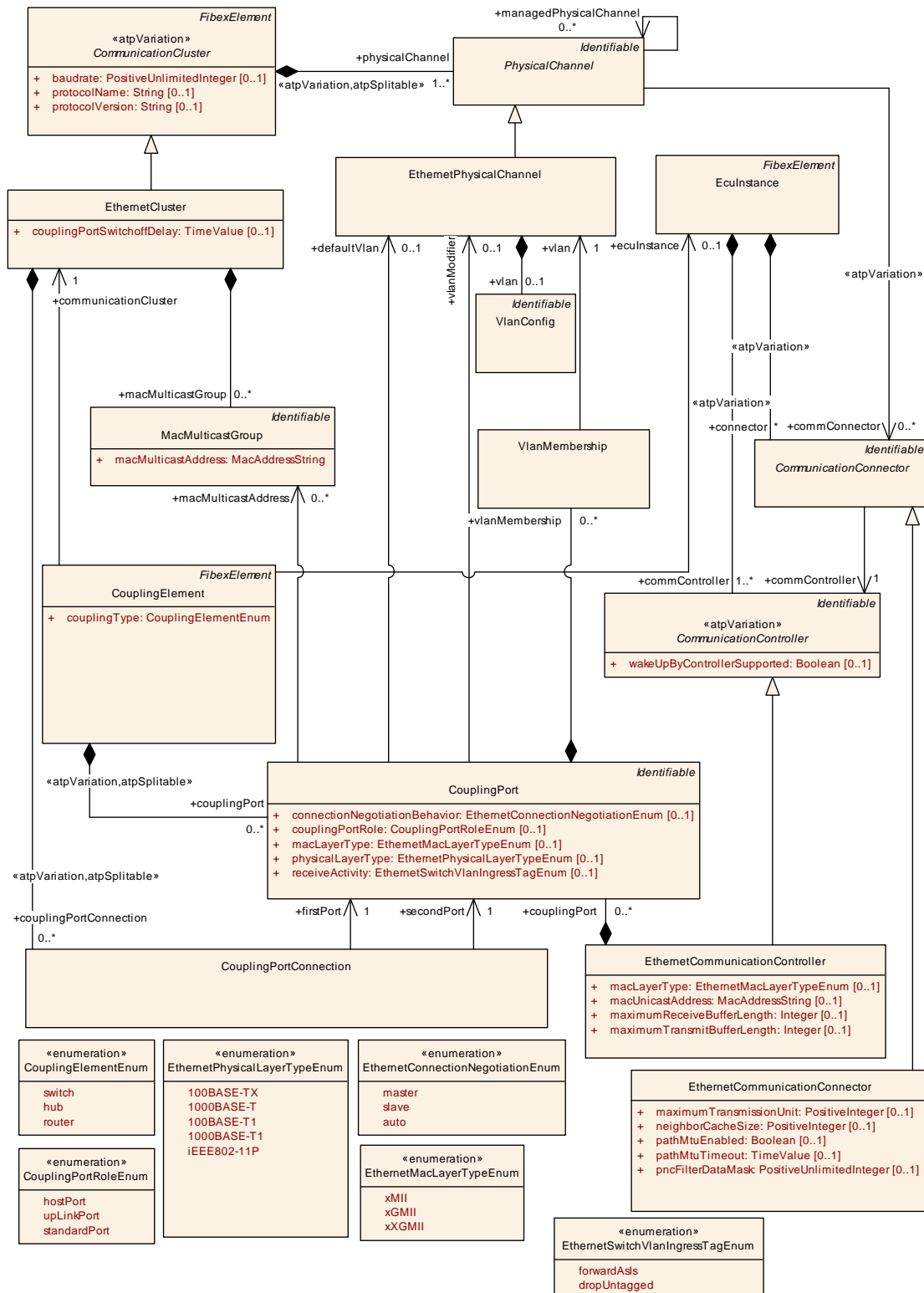


Figure 3.10: Ethernet topology elements (Fibex4Ethernet_Topology)

3.3.6.1 Ethernet Cluster

Each `EthernetCluster` may have globally defined `MacMulticastGroups`. `MacMulticastGroups` have a `macMulticastAddress` (for example 01:00:5E:7F:FF:FF). One sender can handle many receivers simultaneously, if the receivers have all the same `macMulticastAddress`.

[constr_3047] Uniqueness of `macMulticastAddresses` [A `macMulticastAddress` shall be unique in a particular `EthernetCluster`.]()

Class	<<atpVariation>> EthernetCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet-specific cluster attributes. Tags: atp.ManifestKind=MachineManifest atp.recommendedPackage=CommunicationClusters			
Base	<i>ARObject, CollectableElement, CommunicationCluster, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
couplingPort Connection	CouplingPort Connection	*	aggr	Specification of connections between CouplingElements and EculInstances. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=couplingPortConnection, variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingPort SwitchoffDelay	TimeValue	0..1	attr	Switch off delay for CouplingPorts in seconds. It denotes the delay of switching off couplingPorts after the request to switch off a couplingPort was issued. (e.g. switch off of Ethernet switch ports).
macMulticast Group	MacMulticastGroup	*	aggr	MacMulticastGroup that is defined for the Subnet (EthernetCluster).

Table 3.46: EthernetCluster

Class	MacMulticastGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Per EthernetCluster globally defined MacMulticastGroup. One sender can handle many receivers simultaneously if the receivers have all the same macMulticastAddress. The addresses need to be unique for the particular EthernetCluster.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
macMulticast Address	MacAddressString	1	attr	A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.

Table 3.47: MacMulticastGroup

3.3.6.2 Ethernet Physical Channel

The `EthernetPhysicalChannel` represents a VLAN. VLANs (IEEE 802.1q) divide physical Ethernet networks in logical subnets. Their realization requires switches with VLAN support. VLANs are defined on a switch on a port-by-port basis.

The term `EthernetPhysicalChannel` may be misleading because it actually does *not* define the physical (electrical) attributes of the communication but the `EthernetPhysicalChannel` defines the VLANs as *logical* broadcast domains in which the communication partners can interact.

Regardless whether the Ethernet communication uses tagged [`TPS_SYST_01095`] or untagged [`TPS_SYST_01096`] VLANs all communication needs to be defined within respective `EthernetPhysicalChannels` as defined in chapter 6.1.

[`TPS_SYST_01095`] tagged VLANs [In the System Description a VLAN is represented by an `EthernetPhysicalChannel` and is identified by its `vlanIdentifier`.] (`RS_SYST_00039`)

[`TPS_SYST_01096`] untagged VLANs [If the `VlanConfig` and the `vlanIdentifier` are not defined for an `EthernetPhysicalChannel` than the channel is called “untagged”.] (`RS_SYST_00039`)

Every `Frame` that is sent over a “tagged” VLAN is tagged with a VLAN Tag. With this tag every receiving switch has the information about the VLAN that the `Frame` belongs to. The VLAN Tag that is attached to a `Frame` contains the user priority for the `Frame` that is described with the `defaultPriority` and the `vlanIdentifier`.

Class	<code>EthernetPhysicalChannel</code>			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	The <code>EthernetPhysicalChannel</code> represents a VLAN or an untagged channel. An untagged channel is modeled as an <code>EthernetPhysicalChannel</code> without an aggregated VLAN. Tags: atp.ManifestKind=MachineManifest			
Base	<code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PhysicalChannel</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
network Endpoint	<code>NetworkEndpoint</code>	*	aggr	Collection of <code>NetworkEndpoints</code> that are used in the VLAN. Stereotypes: atpSplittable Tags: atp.Splitkey=shortName
soAdConfig	<code>SoAdConfig</code>	0..1	aggr	SoAd Configuration for one specific Physical Channel.
vlan	<code>VlanConfig</code>	0..1	aggr	VLAN Configuration.

Table 3.48: EthernetPhysicalChannel

[`constr_3333`] Standardized values for the attribute `category` of meta-class `EthernetPhysicalChannel` [The following values of the attribute `category` of meta-class `EthernetPhysicalChannel` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetPhysicalChannel` in case of a wired ethernet connection

- WIRELESS: This represents the usage of the `EthernetPhysicalChannel` in case of a wireless ethernet connection

]()

[TPS_SYST_02159] Default value for the attribute `category` of meta-class `EthernetPhysicalChannel` [The default value for the `category` of an `EthernetPhysicalChannel` shall be `WIRED`.]()

[constr_3334] Allowed references between `EthernetPhysicalChannel` and `EthernetCommunicationConnector` [An `EthernetPhysicalChannel` is only allowed to reference `EthernetCommunicationConnectors` in the role `commConnector` that have the same `category` value as the referencing `EthernetPhysicalChannel`.]()

[constr_3365] `EthernetPhysicalChannels` with different `category` values are not allowed within an `EthernetCluster` [A mix of `EthernetPhysicalChannels` with different `category` values within an `EthernetCluster` is currently not supported by AUTOSAR.]()

[constr_3336] `EthernetPhysicalChannel.soAdConfig` in case of WIRELESS `EthernetPhysicalChannel` [If `EthernetPhysicalChannel` has the `category` `WIRELESS` then the `EthernetPhysicalChannel` shall not aggregate the `SoAdConfig`.]()

[TPS_SYST_01086] Number of Ethernet channels [Each `EthernetCluster` may aggregate up to 4096 `EthernetPhysicalChannels`.] (*RS_SYST_00039*)

Class	VlanConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	VLAN Configuration attributes			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
vlanIdentifier	PositiveInteger	1	attr	A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.

Table 3.49: VlanConfig

[constr_3048] Range of `vlanIdentifier` [The allowed values of `vlanIdentifier` range from 0 to 4095.]()

3.3.6.2.1 VLAN Priority

The Priority is a 3-bit field which refers to the IEEE 802.1Q priority. It indicates the frame priority level. Values are from 0 (best effort) to 7 (highest); 1 represents the lowest priority. These values can be used to prioritize different classes of traffic (voice, video, data, etc.). The priority is contained in the Ethernet Header together with the `vlanIdentifier`.

The `defaultPriority` can be overwritten on different levels:

1. `NetworkEndpoint`
2. `ApplicationEndpoint`
3. `ProvidedServiceInstance` or `ConsumedEventGroup`

If a priority on an `ApplicationEndpoint` is defined the priorities in the `NetworkEndpoint` and the `defaultPriority` in the `VlanMembership` would be ignored.

The following table shows two `CouplingPorts`. Both have two `NetworkEndpoints` and for each `NetworkEndpoint` two `ApplicationEndpoints` are defined. This means that per Port two IP Addresses and four Tcp-Ports are used. On each level a priority may be defined.

For NEP1.1 no priority is defined. This means that the Default-Priority from Coupling-Port1 is valid. On CouplingPort1 all messages have the Priority 0 ("best effort") except for messages that are going over `ApplicationEndpoint` AEP1.1.2 and AEP 1.2.2. These messages have the priority 1 (higher priority). On CouplingPort2 the priority is overwritten on several levels. Please note that AEP 2.2.1 and AEP 2.2.2 are reducing the priority that is defined on the NEP2.2.

Port (Default-Prio)	NetworkEndpoint (e.g. IpAddress)	ApplicationEndpoint (e.g. Tcp Port)
CouplingPort1: Prio.0	NEP1.1: Prio. —	AEP 1.1.1: Prio. —
		AEP 1.1.2: Prio. 1
	NEP1.2: Prio. 0	AEP 1.2.1: Prio. —
		AEP 1.2.2: Prio. 1
CouplingPort2: Prio.0	NEP2.1: Prio. 1	AEP 2.1.1: Prio. 2
		AEP 2.1.2: Prio. 3
	NEP2.2: Prio. 2	AEP 2.2.1: Prio. 1
		AEP 2.2.2: Prio. 0

Table 3.50: VLAN Priority Example

3.3.6.3 Ethernet Coupling Elements and Coupling Ports

A `CouplingElement` is used to connect `EcuInstances` via `CouplingPorts` to `EthernetPhysicalChannels` (VLANs) that are defined within an `EthernetCluster`.

`CouplingElements` can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A `CouplingElement` references the `EthernetCluster` and contains a collection of available `CouplingPorts`. The `couplingType` identifies the `CouplingElement` as a switch, hub or router.

Class	CouplingElement			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. Coupling Elements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device. Tags: atp.recommendedPackage=CouplingElements			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
communicationCluster	EthernetCluster	1	ref	This relationship defines to which cluster the Coupling Element belongs.
couplingPort	CouplingPort	*	aggr	Hardware Port of the CouplingElement that is used to connect this CouplingPort to EcuInstances or other CouplingElements. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingType	CouplingElementEnum	1	attr	Describes the coupling type of this CouplingElement.
ecuInstance	EcuInstance	0..1	ref	Optional reference to the ECU where the Coupling Element is located.

Table 3.51: CouplingElement

Enumeration	CouplingElementEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Identifies the Coupling type.
Literal	Description
hub	A device that is used to connect segments of a LAN. In Hubs frames are "broadcasted" to every one of its ports. Tags: atp.EnumerationLiteralIndex=0
router	A device that routes frames between different networks. Tags: atp.EnumerationLiteralIndex=1
switch	A device that filters and forwards frames between different LAN segments. Tags: atp.EnumerationLiteralIndex=2

Table 3.52: CouplingElementEnum

[constr_3062] The **EcuInstance** that is referenced from a specific **CouplingElement** shall be connected to the same **EthernetCluster** as the specific **CouplingElement** [The **EcuInstance** referenced from a specific **CouplingElement** in the role **ecuInstance** shall be connected via the **CommunicationConnector** and a **EthernetPhysicalChannel** that refers the **CommunicationConnector** to the **EthernetCluster** referenced by the specific **CouplingElement** in the role **communicationCluster**.]()

Class	CouplingPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
connection Negotiation Behavior	EthernetConnectionNegotiationEnum	0..1	attr	Specifies the connection negotiation of the CouplingPort. Tags: atp.Status=shallBecomeMandatory
couplingPort Details	CouplingPortDetails	0..1	aggr	Defines more details of a CouplingPort in case a more specific configuration is required.
couplingPort Role	CouplingPortRoleEnum	0..1	attr	Defines the role this CouplingPort takes in the context of the CouplingElement.
defaultVlan	EthernetPhysicalChannel	0..1	ref	The vLanIdentifier of the referenced VLAN is the Default-PVID (port VLAN ID). A Port VLAN ID is a default VLAN ID that is assigned to an access CouplingPort to designate the VLAN segment to which this port is connected. Also, if a CouplingPort has not been configured with any VLAN memberships, the virtual switch's Port VLAN ID (pvid) becomes the default VLAN ID for the ports connection. This identifier/tag is added for incoming untagged messages at the port (ingress tagging). For outgoing messages with this identifier, the tag is removed at the port (egress untagging, depending on the Vlan Membership.sendActivity).
macLayerType	EthernetMacLayerTypeEnum	0..1	attr	Specifies the mac layer type of the CouplingPort.
macMulticast Address	MacMulticastGroup	*	ref	Assigns a set of MAC-Multicast-Addresses which are addressable via this CouplingPort. This is a static pre-configuration and further addresses may be learned during runtime.
physicalLayer Type	EthernetPhysicalLayerTypeEnum	0..1	attr	Specifies the physical layer type of the CouplingPort.
pncMapping	PncMappingIdent	*	ref	Reference to the partial networks this CouplingPort participates in.
receiveActivity	EthernetSwitchVlanIngressTagEnum	0..1	attr	Defines the handling of frames at the ingress port.
vlan Membership	VlanMembership	*	aggr	Messages of VLANs that are defined here can be communicated via the CouplingPort.
vlanModifier	EthernetPhysicalChannel	0..1	ref	All incoming messages at this CouplingPort shall be tagged with this VLAN Id. This tagging is performed regardless whether the message already has a VLAN tag or is untagged, an existing VLAN tag will be overwritten. This feature is XOR with CoupligPort.defaultVlan.

Table 3.53: CouplingPort

Enumeration	EthernetConnectionNegotiationEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies connection negotiation types of Ethernet transceiver links.
Literal	Description
auto	Automatic Negotiation Tags: atp.EnumerationLiteralIndex=0
master	Master Tags: atp.EnumerationLiteralIndex=1
slave	Slave Tags: atp.EnumerationLiteralIndex=2

Table 3.54: EthernetConnectionNegotiationEnum

Enumeration	EthernetMacLayerTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies MAC (Media Access Control) Layer types.
Literal	Description
xGMII	Mac layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII) Tags: atp.EnumerationLiteralIndex=1 xml.name=XG-MII
xMII	Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII) Tags: atp.EnumerationLiteralIndex=0 xml.name=X-MII
xxGMII	Mac layer interface (data) bandwidth class 10Gbit/s Tags: atp.EnumerationLiteralIndex=2 xml.name=XXG-MII

Table 3.55: EthernetMacLayerTypeEnum

Enumeration	EthernetPhysicalLayerTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies physical layer types of Ethernet transceiver links.
Literal	Description
_1000BASE_T	Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs. Tags: atp.EnumerationLiteralIndex=6 xml.name=1000BASE-T
_1000BASE_T1	Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable. Tags: atp.EnumerationLiteralIndex=8 xml.name=1000BASE-T1
_100BASE_T1	Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100BASE-T1 is the IEEE Standardized version of BroadRReach. Tags: atp.EnumerationLiteralIndex=7 xml.name=100BASE-T1





Enumeration	EthernetPhysicalLayerTypeEnum
_100BASE_TX	Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs. Tags: atp.EnumerationLiteralIndex=5 xml.name=100BASE-TX
IEEE802_11P	Ethernet Standard (IEEE 802.11p) to support wireless communication in vehicular environments. Tags: atp.EnumerationLiteralIndex=9 xml.name=IEEE802-11P

Table 3.56: EthernetPhysicalLayerTypeEnum

Enumeration	EthernetSwitchVlanIngressTagEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines the possible tagging behavior at an ingress port.
Literal	Description
dropUntagged	Drop if untagged. Tags: atp.EnumerationLiteralIndex=1
forwardAsIs	Forward with the same VLAN as received. Also untagged frames will be forwarded as untagged. Tags: atp.EnumerationLiteralIndex=0

Table 3.57: EthernetSwitchVlanIngressTagEnum

Class	VlanMembership			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Static logical channel or VLAN binding to a switch-port. The reference to an EthernetPhysicalChannel without a VLAN defined represents the handling of untagged frames.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
defaultPriority	PositiveInteger	1	attr	Standard output-priority outgoing Frames will be tagged with. Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed. In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.
dhcpAddress Assignment	DhcpServer Configuration	0..1	aggr	Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddress Assignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.





Class	VlanMembership			
sendActivity	EthernetSwitchVlanEgressTaggingEnum	0..1	attr	Attribute denotes whether a VLAN tagged ethernet frame will be <ol style="list-style-type: none"> 1. sent with its VLAN tag (sentTagged) 2. sent without a VLAN tag (sentUntagged) 3. will be dropped at this port (notSent or VLAN not member of this list)
vlan	EthernetPhysicalChannel	1	ref	References a channel that represents a VLAN or an untagged channel.

Table 3.58: VlanMembership

Class	CouplingPortConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Connection between two CouplingPorts (firstPort and secondPort).			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
firstPort	CouplingPort	1	ref	Reference to the first CouplingPort that is connected via the CouplingPortConnection.
secondPort	CouplingPort	1	ref	Reference to the second CouplingPort that is connected via the CouplingPortConnection.

Table 3.59: CouplingPortConnection

[CouplingPorts](#) are hardware ports of [CouplingElements](#) and [EcuInstances](#). Connections between [CouplingPorts](#) are realized through [CouplingPortConnections](#).

Optionally the [CouplingPort](#) of a [CouplingElement](#) may also have one or several [VlanMemberships](#), a [defaultVlan](#) reference and a reference to a [MacMulticastGroup](#).

[constr_3521] defaultVlan and vlanMembership [If a [CouplingPort](#) refers to an [EthernetPhysicalChannel](#) in the role [defaultVlan](#) the [CouplingPort](#) shall also have a [vlanMembership](#) defined. This [VlanMembership](#) shall point to the same [EthernetPhysicalChannel](#) in the role [vlan](#) as the [defaultVlan](#).]()

[constr_3522] vlanModifier and vlanMembership [If a [CouplingPort](#) refers to an [EthernetPhysicalChannel](#) in the role [vlanModifier](#) the [CouplingPort](#) shall also have a [vlanMembership](#) defined. This [VlanMembership](#) shall point to the same [EthernetPhysicalChannel](#) in the role [vlan](#) as the [vlanModifier](#).]()

[constr_3435] Applicability of CouplingPort.macMulticastAddress [The reference [CouplingPort.macMulticastAddress](#) is only applicable if the [CouplingPort](#) is aggregated by a [CouplingElement](#) with [couplingType](#) = [switch](#).]()

[constr_3133] physicalLayerType of connected CouplingPorts [The [physicalLayerType](#) of two [CouplingPorts](#) which are connected via a [CouplingPortConnection](#) shall be equal.]()

[constr_3134] The connection of two `CouplingPorts` with `connectionNegotiationBehavior` set to `master` is forbidden [The `connectionNegotiationBehavior` of two `CouplingPorts` which are connected via a `CouplingPortConnection` shall not be both set to `master`.]()

[constr_3135] The connection of two `CouplingPorts` with `connectionNegotiationBehavior` set to `slave` is forbidden [The `connectionNegotiationBehavior` of two `CouplingPorts` which are connected via a `CouplingPortConnection` shall not be both set to `slave`.]()

[TPS_SYST_01097] Assignment of `CouplingPorts` to a VLAN [`CouplingPorts` of `CouplingElements` can be assigned to VLANs (`EthernetPhysicalChannels`) with the `vlanMembership` aggregation.](*RS_SYST_00039*)

[TPS_SYST_01098] Assignment of `CouplingPorts` to an “untagged” VLAN [A `CouplingPort` may be assigned to several VLANs, but only one of those assignments can be “untagged”.](*RS_SYST_00039*)

[constr_3534] `EthernetPhysicalChannel` shall only be referenced by one `VlanMembership` [An `EthernetPhysicalChannel` shall only be referenced by one `VlanMembership` in the role `VlanMembership.vlan` in the scope of one `CouplingPort`.]()

Figure 3.11 shows a `CouplingElement` with two `CouplingPorts`.

In this example Port 0 is assigned to three VLANs and one “untagged” `EthernetPhysicalChannel`. VLAN3 is marked as the `defaultVlan`. With the combination of the `defaultVlan` and the `VlanMembership` to the “untagged” `EthernetPhysicalChannel` the `Frames` that are transmitted over Port 0 on VLAN3 are “untagged” on the wire in both directions (Tx and Rx). The switch adds the tag for incoming untagged messages at the port (ingress tagging) and for outgoing messages the tag is removed at the port (egress untagging).

Port 1 is assigned to three VLANs. But the `VlanMembership` to the “untagged” `EthernetPhysicalChannel` is not defined here. For this reason, `Frames` that are transmitted over Port 1 on VLAN3 are “tagged”.

If a `defaultVlan` is defined for a `CouplingPort` but the `defaultVlan` is not referenced by the `VlanMembership` then “untagged” `Frames` can be received via the `CouplingPort`. But a response can not be send back.

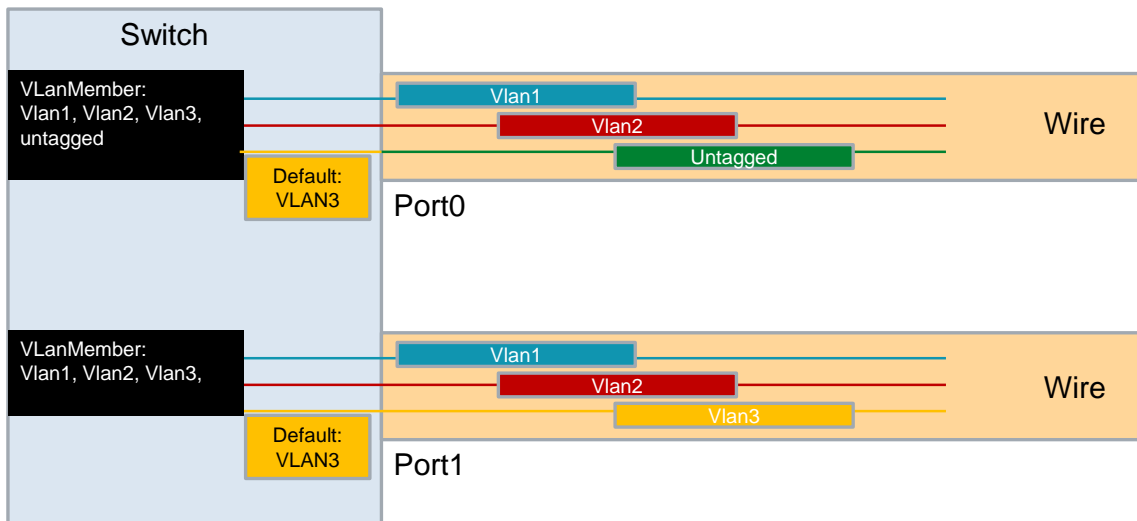


Figure 3.11: Default Vlan Example

3.3.6.4 Ethernet Communication Controller

`EthernetCommunicationController` is a specialization of the `CommunicationController` class. It contains the specific Ethernet controller attributes needed for configuring an `EcuInstance` connected to a certain Ethernet cluster.

Class	<<atpVariation>> EthernetCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific communication port attributes.			
Base	<code>ARObject</code> , <code>CommunicationController</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
couplingPort	<code>CouplingPort</code>	*	aggr	Optional CouplingPort that can be used to connect the ECU to a CouplingElement (e.g. a switch).
macLayerType	<code>EthernetMacLayerTypeEnum</code>	0..1	attr	Specifies the mac layer type of the ethernet controller.
macUnicast Address	<code>MacAddressString</code>	0..1	attr	Media Access Control address (MAC address) that uniquely identifies each EthernetCommunicationController in the network.
maximum ReceiveBuffer Length	Integer	0..1	attr	Determines the maximum receive buffer length (frame length) in bytes.
maximum TransmitBuffer Length	Integer	0..1	attr	Determines the maximum transmit buffer length (frame length) in bytes.

Table 3.60: EthernetCommunicationController

[constr_3535] `EthernetCommunicationController` shall aggregate at most one `CouplingPort` [An `EthernetCommunicationController` is allowed to aggregate at most one `CouplingPort`.]()

[constr_3332] Standardized values for the attribute `category` of meta-class `EthernetCommunicationController` [The following values of the attribute `category` of meta-class `EthernetCommunicationController` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationController` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationController` in case of a wireless ethernet connection

]()

[TPS_SYST_02158] Default value for the attribute `category` of meta-class `EthernetCommunicationController` [The default value for the `category` of an `EthernetCommunicationController` shall be `WIRED`.]()

The `EthernetCommunicationController` has the additional information of a `macUnicastAddress`. This is a globally unique MAC-address for the `CommunicationController`.

3.3.6.5 Ethernet Communication Connector

`EthernetCommunicationConnector` adds the Ethernet specific attributes to the `CommunicationConnector`.

Class	EthernetCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific attributes to the <code>CommunicationConnector</code> . Tags: atp.ManifestKind=MachineManifest			
Base	<code>ARObject</code> , <code>CommunicationConnector</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
maximumTransmissionUnit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
neighborCacheSize	PositiveInteger	0..1	attr	This attribute specifies the size of neighbor cache or ARP table in units of entries.
networkEndpoint	<code>NetworkEndpoint</code>	*	ref	NetworkEndpoints
pathMtuEnabled	Boolean	0..1	attr	If enabled the IPv4/IPv6 processes incoming ICMP "Packet Too Big" messages and stores a MTU value for each destination address.
pathMtuTimeout	TimeValue	0..1	attr	If this value is >0 the IPv4/IPv6 will reset the MTU value stored for each destination after n seconds.
pncFilterDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for Ethernet Payload used to configure the Ethernet Transceiver for partial network wakeup.

Table 3.61: EthernetCommunicationConnector

[TPS_SYST_02166] Derivation of UdpNmPnFilterMaskByte [The `UdpNmPnFilterMaskByte` should not be computed from the `pncIdentifier` values in order to support future introduction of additional PNCs.

Note that for one `EcuInstance` all contributing `EthernetCommunicationConnector.pncFilterDataMask` will be bitwise ORed to obtain the value of `UdpNmPnFilterMaskByte`. Note that this data mask is calculated over the whole payload of the `NmPdu` ignoring the leading bytes which do not contain `pncVector` information. The number of leading bytes which shall be ignored is equivalent to the value of `System.pncVectorOffset.` (*RS_SYST_00042*)

[constr_3331] Standardized values for the attribute `category` of meta-class `EthernetCommunicationConnector` [The following values of the attribute `category` of meta-class `EthernetCommunicationConnector` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationConnector` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationConnector` in case of a wireless ethernet connection

]()

[TPS_SYST_02157] Default value for the attribute `category` of meta-class `EthernetCommunicationConnector` [The default value for the `category` of an `EthernetCommunicationConnector` shall be *WIRED.*]()

[constr_3335] Allowed references between `EthernetCommunicationConnector` and `EthernetCommunicationController` [An `EthernetCommunicationConnector` is only allowed to reference an `EthernetCommunicationController` in the role `commController` that has the same `category` value as the referencing `EthernetCommunicationConnector.`]()

3.3.6.6 Ethernet Switch Driver

Ethernet networks in an automotive environment consist basically of ECUs with a single port PHY and switch ECUs with several ports. Different to consumer networks, where switches are typically stand-alone devices, switches in automotive networks may be integrated and connected to a CPU via MII and other interfaces. The configuration of these switches does influence the communication behavior within the network.

3.3.6.6.1 Ethernet switch port structure

In order to describe switched Ethernet networks it is essential to describe some parts of an Ethernet switch. Examples are scheduling and forwarding mechanisms within a switch as well as the switch structure within its ports.

As shown in figure 3.12, the switch consists of a certain number of ports. Each port has its own set of egress FIFOs in which the incoming packets are buffered. How the messages in the FIFOs will be forwarded depends mainly on the shaping and port scheduling mechanisms. Thus, the parametrization of the egress port influences the latency of messages within the network.

Please note that the egress port structures in figure 3.12 are meant as an example. Other structures with different FIFO numbers are possible as well.

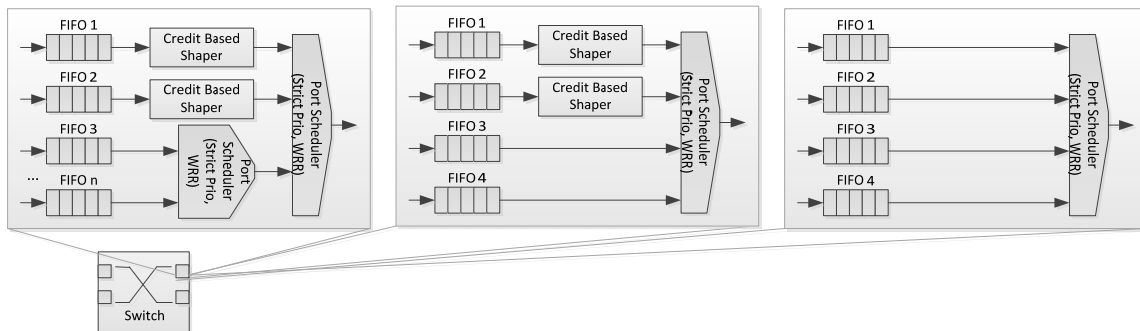


Figure 3.12: Example egress switch port configurations

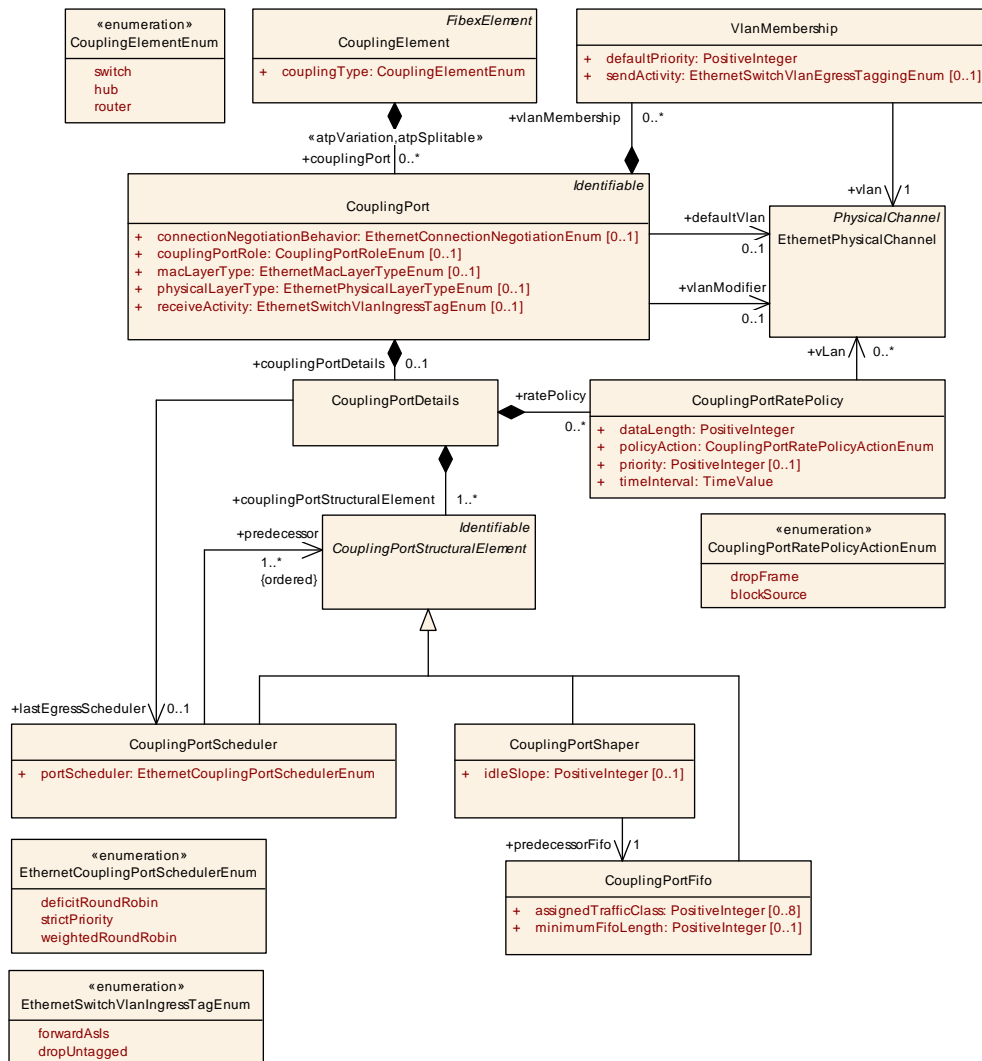


Figure 3.13: Egress switch port structure

The structural description of an Ethernet switch is based on the already existing `CouplingElement` in the System Template. Each `CouplingElement` can already have a set of `CouplingPorts`.

In case a detailed Switch configuration is required, there is the configuration option to add to the `CouplingPort` a `CouplingPortDetails` element which encapsulates the structural description of one switch port.

The elements which one switch port consists of are (egress side):

- `CouplingPortFifo`
- `CouplingPortShaper`
- `CouplingPortScheduler`

The model allows to collect the egress parts of one switch port in the `CouplingPortDetails.couplingPortStructuralElements`.

[TPS_SYST_03006] Ethernet switch egress port setup [Two setups can be defined at an egress port of a switch:

- The switch port has only one Fifo:
 - the `CouplingPortFifo` element is aggregated at the `CouplingPortDetails.couplingPortStructuralElements`
 - no `CouplingPortDetails.lastEgressScheduler` is defined.
- The switch port has at least one scheduler
 - the various switch port elements are all aggregated at the `CouplingPortDetails.couplingPortStructuralElements`
 - the `CouplingPortScheduler` which is the last scheduler in a chain of structural elements is additionally referenced in the role `CouplingPortDetails.lastEgressScheduler`

]([RS_SYST_00052](#))

The modeling approach is based on a predecessor chain model where the chain is started by the last scheduler in the switch port and defines where the input to this scheduler comes from. The input to a scheduler can come from several predecessor elements which might be

- another `CouplingPortScheduler`
- a `CouplingPortShaper`
- a `CouplingPortFifo`.

[TPS_SYST_03007] Ethernet port scheduler algorithm [The scheduler performs a prioritization of the incoming frames based on the algorithm defined in the `CouplingPortScheduler.portScheduler`.]([RS_SYST_00052](#))

[TPS_SYST_03008] Ethernet port scheduler priority [The first element in `CouplingPortScheduler.predecessor` has the highest priority. Therefore, it is important to have the predecessor definition of the scheduler ordered.]([RS_SYST_00052](#))

Another restriction is that a `CouplingPortShaper` can only have a `CouplingPortFifo` as `predecessorFifo`, which is given by the model.

[TPS_SYST_03009] Ethernet port shaper `idleSlope` [The `idleSlope` is defined in the IEEE802.1Qav standard as a parameter for an increase of credit in bits per second. The `idleSlope` can never exceed the maximal transmit rate of a port, e.g. 100Mbits for BroadR-Reach and 1Gbits for RTPGE. The `idleSlope` determines the maximum fraction of the port transmit rate that is available for the queue associated with the shaper: $\text{bandwidthFraction} = \text{idleSlope} / \text{portTransmitRate}$.]([RS_SYST_00052](#))

Class	CouplingPortDetails			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines details of a CouplingPort. May be used to configure the structures of a switch.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
couplingPort Structural Element	CouplingPortStructuralElement	1..*	aggr	Collects all the structural parts at which a CouplingPort may be configurable.
ethernetPriority Regeneration	EthernetPriorityRegeneration	0..8	aggr	Defines a priority regeneration where the ingress priority is replaced by regenerated priority.
ethernetTraffic Class Assignment	CouplingPortTrafficClassAssignment	0..8	aggr	Defines the ingress port to EthernetTrafficClass assignment.
globalTime Props	GlobalTimeCouplingPortProps	0..1	aggr	Specifies properties for the usage of the CouplingPort in the scope of Global Time Sync.
lastEgress Scheduler	CouplingPortScheduler	0..1	ref	Defines which CouplingPortScheduler is the last in the egress port structure.
ratePolicy	CouplingPortRatePolicy	*	aggr	Rate policies to be applied for this CouplingPort.

Table 3.62: CouplingPortDetails

Class	<i>CouplingPortStructuralElement</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	General class to define structural elements a CouplingPort may consist of.			
Base	<i>ARObject</i> , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	CouplingPortFifo , CouplingPortScheduler , CouplingPortShaper			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.63: CouplingPortStructuralElement

Class	CouplingPortScheduler			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a scheduler for the CouplingPort egress structure.			
Base	<i>ARObject</i> , <i>CouplingPortStructuralElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
portScheduler	EthernetCouplingPortSchedulerEnum	1	attr	Defines the schedule algorithm to be used.
predecessor (ordered)	CouplingPortStructuralElement	1..*	ref	Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.

Table 3.64: CouplingPortScheduler

Enumeration	EthernetCouplingPortSchedulerEnum			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the schedule algorithm to be used.			
Literal	Description			
deficitRoundRobin	Schedule algorithm "deficit round robin" Tags: atp.EnumerationLiteralIndex=0			
strictPriority	Schedule algorithm "strict priority" Tags: atp.EnumerationLiteralIndex=1			
weightedRoundRobin	Schedule algorithm "weighted round robin" Tags: atp.EnumerationLiteralIndex=2			

Table 3.65: EthernetCouplingPortSchedulerEnum

Class	CouplingPortShaper			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a shaper for the CouplingPort egress structure.			
Base	<i>ARObject</i> , <i>CouplingPortStructuralElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
idleSlope	PositiveInteger	0..1	attr	Defines the increase of credit in bits per second for the AVB shaper.
predecessorFifo	CouplingPortFifo	1	ref	Defines the CouplingPortFifo which provides the input to this shaper.

Table 3.66: CouplingPortShaper

Class	CouplingPortFifo			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a Fifo for the CouplingPort egress structure.			
Base	<i>ARObject</i> , <i>CouplingPortStructuralElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note





Class	CouplingPortFifo			
assignedTrafficClass	PositiveInteger	0..8	attr	Defines a set of Traffic Classes which shall be handled by this Fifo. range: 0-7
minimumFifoLength	PositiveInteger	0..1	attr	FIFO minimum length in Byte. An actual configuration/hardware may use a bigger value.

Table 3.67: CouplingPortFifo

3.3.6.6.2 Ethernet switch rate policy

A [CouplingPort](#) may define a [CouplingPortRatePolicy](#) via the [CouplingPortDetails.ratePolicy](#).

Class	CouplingPortRatePolicy			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a rate policy on a CouplingPort.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
dataLength	PositiveInteger	1	attr	Amount of data in bytes (excluding header information) that can be received to define the rate policy.
policyAction	CouplingPortRatePolicyActionEnum	1	attr	Defines the action to be performed when this rate policy is violated.
priority	PositiveInteger	0..1	attr	Defines the priority which this rate policy shall be limited on. If no priority is given this rate policy is not considering priority.
timeInterval	TimeValue	1	attr	Time interval used to define the base of the rate policy.
vLan	EthernetPhysicalChannel	*	ref	Defines the VLANs this rate policy shall be limited on. If no VLAN is given this rate policy is not considering VLAN tags.

Table 3.68: CouplingPortRatePolicy

Enumeration	CouplingPortRatePolicyActionEnum	
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology	
Note	Defines the action to be performed when a rate policy is violated.	
Literal	Description	
blockSource	If the rate policy is violated the CouplingPort this CouplingPortRatePolicy is defined on shall block all frames from the MAC-Address the violation was caused by. Tags: atp.EnumerationLiteralIndex=1	
dropFrame	If the rate policy is violated the frame shall be dropped. Tags: atp.EnumerationLiteralIndex=0	

Table 3.69: CouplingPortRatePolicyActionEnum

3.3.6.6.3 Ethernet packet forwarding

Besides the modeling of egress ports, it is necessary to specify how incoming packets are forwarded to the egress ports. For this purpose, different assignment policies of packets to egress port FIFOs are implemented in switches.

As an example, the Ethernet priority field can be evaluated and remapped into a regenerated priority: Within the VLAN-tag, the PCP-field (priority code point) is a parameter which can be modified at an ingress port of an Ethernet switch. For this purpose a priority regeneration table can be defined.

The `CouplingPortDetails.ethernetPriorityRegeneration` is optional in case the feature of priority regeneration is not be used.

[TPS_SYST_03003] Ethernet priority regeneration [The `CouplingPortDetails.ethernetPriorityRegeneration` specifies which `ingressPriority` is mapped to which `regeneratedPriority`.] (*RS_SYST_00052*)

[constr_3515] Fully filled EthernetPriorityRegeneration table [In case the `CouplingPortDetails.ethernetPriorityRegeneration` is defined it shall contain exactly 8 elements of `EthernetPriorityRegeneration`, one for each value of `ingressPriority` (0-7).] ()

The (potentially remapped) Ethernet priority field can be evaluated and mapped to a traffic class. Such a traffic class is again mapped to an egress FIFO. Other header information of the Ethernet frame can be also used for the assignment of Ethernet frames to egress FIFOs. For the mapping to a certain traffic class, the following tables are necessary.

PORT-based Mapping	Traffic Class
Port2, Port3, Port4	7
Port1	6
–	5
–	4
–	3
–	2
–	1
–	0

Table 3.70: Port to Traffic Class mapping

PCP-based Mapping	Traffic Class
Prio 0	7
Prio 1	6
Prio 2-7	5
–	4
–	3
–	2
–	1
–	0

Table 3.71: PCP-field to Traffic Class mapping

While the first table shows the mapping of ingress-ports to traffic classes, the second table shows the priority-based mapping which can be defined per ingress port. Both tables are in conflict with each other, i.e. it has to be decided which mapping is applied.

Also the mapping of a traffic class to a FIFO shall be done on a per port basis. An example is shown in the following table.

Traffic Class	FIFO (if 4 FIFOs available)
7	3
6	2
0-5	1
–	0

Table 3.72: Traffic Class to FIFO mapping

In order to model the relationship between the ingress port and the egress port, the `CouplingPortTrafficClassAssignment` elements are used.

[TPS_SYST_03010] Ethernet switch packet to traffic class assignment [First the ingress packets are assigned to traffic classes. The two use-cases from above are both supported by this model:

- Port to traffic class mapping (only one traffic class per port possible) from table 3.70
 - `CouplingPortDetails` has exactly one `ethernetTrafficClassAssignment` defined
 - the `CouplingPortTrafficClassAssignment` has no `priority` defined
- PCP-field to Traffic Class Mapping from table 3.71
 - for each traffic class the `CouplingPortDetails` aggregate one `ethernetTrafficClassAssignment`
 - each `CouplingPortTrafficClassAssignment` element has a set of `priority`s defined which shall be mapped to the given `trafficClass`

](*RS_SYST_00052*)

[constr_5049] Ethernet switch packet to traffic class assignment restriction [For one `CouplingPortDetails` there exists either

- one `ethernetTrafficClassAssignment` with no `priority` attribute or
- up to 8 `ethernetTrafficClassAssignment` elements with a set of `priority` attributes

]()

[TPS_SYST_03011] Ethernet switch traffic class to FIFO assignment [Second, the traffic classes are assigned to the switch egress FIFOs. The `CouplingPortFifo` has

a set of `assignedTrafficClass` elements. These defined traffic classes shall be forwarded to this FIFO.] (*RS_SYST_00052*)

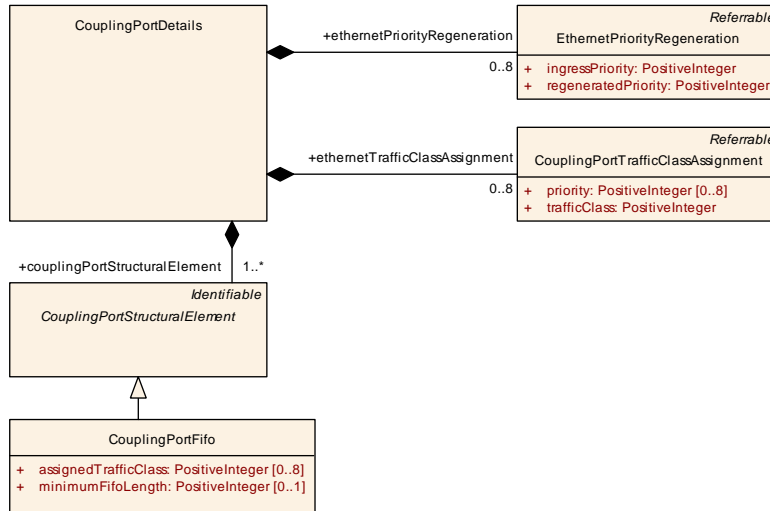


Figure 3.14: Ethernet Priority Regeneration and Ethernet Traffic Class Assignment

Class	EthernetPriorityRegeneration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a priority regeneration where the ingressPriority is replaced by regeneratedPriority. The ethernetPriorityRegeneration is optional in case no priority regeneration shall be performed. In case a ethernetPriorityRegeneration is defined it shall have 8 mappings, one for each priority.			
Base	ARObject, Referrable			
Attribute	Type	Mult.	Kind	Note
ingressPriority	PositiveInteger	1	attr	Message priority of the incoming message. range: 0-7
regeneratedPriority	PositiveInteger	1	attr	Regenerated message priority. range: 0-7

Table 3.73: EthernetPriorityRegeneration

Class	CouplingPortTrafficClassAssignment			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the assignment of Traffic Class to a frame.			
Base	ARObject, Referrable			
Attribute	Type	Mult.	Kind	Note
priority	PositiveInteger	0..8	attr	Defines a priority which is mapped onto a Traffic Class.
trafficClass	PositiveInteger	1	attr	Defines the Traffic Class which is assigned. range: 0-7

Table 3.74: CouplingPortTrafficClassAssignment

3.3.6.6.4 Ethernet VLAN Configuration

For each VLAN identifier a table is necessary which stores at which egress port the corresponding VLAN is tagged or untagged. For an 8-port switch, this table could look like the following example where T stands for tagging and U for untagging:

VLAN-Id	Port number							
	1	2	3	4	5	6	7	8
1	T	T	-	U	-	-	-	T
2	T	U	-	T	-	-	-	T
...								
4094								

Table 3.75: VLAN Forwarding table

Incoming packets which contain a VLAN-ID of e.g. 1 can be forwarded to the ports 1, 2, 4, and 8. At ports 1, 2, and 8 these packets will be transmitted with the VLAN tag and at port 4 the tag will be removed. If a broadcast message with e.g. VLAN-ID 2 will be received at port 2 it will be forwarded to port 1, 4, and 8. The other ports 3, 5, 6, and 7 are not in the same VLAN. Thus, the packet will not be forwarded to these egress ports. The table considers only messages which contain a VLAN-ID within the switch.

`CouplingPort.vlanMembership` defines specific attributes to the behavior a packet with a specific VLAN-ID shall have on this `CouplingPort`.

[TPS_SYST_03004] VLAN specific sending behavior [The `VlanMembership.sendActivity` defines for a `CouplingPort` and VLAN the sending behavior:

- `sentTagged`: packet is sent at this `CouplingPort` with the defined VLAN-ID
- `sentUntagged`: packet is sent at this `CouplingPort` but the VLAN-ID is removed before sending
- `notSent`: packet is not sent at this `CouplingPort`

](*RS_SYST_00052*)

Another table specifies a port-based modification of the VLAN-ID or an insertion of the VLAN-ID into the Ethernet message:

Port number	1	2	3	4	5	6	7	8
VLAN-Id	2	-	-	6	-	-	-	-

Table 3.76: Ingress VLAN Modification/Insertion Table

In this example, all incoming messages at port one will get the VLAN-Id 2 no matter whether they already had one before. At port 4, all incoming messages will get a 6 as their VLAN-Id. At the remaining ports, no VLAN-Ids will be inserted and an existing VLAN-Id in the Ethernet-message will remain without modification.

[TPS_SYST_03005] **VLAN re-tagging** [All incoming messages at a `CouplingPort` where the `CouplingPort.vlanModifier` is defined shall be tagged with the VLAN-Id defined in `CouplingPort.vlanModifier`. This tagging is performed regardless whether the message already has a VLAN tag or is untagged, an existing VLAN tag shall be overwritten.] (*RS_SYST_00052*)

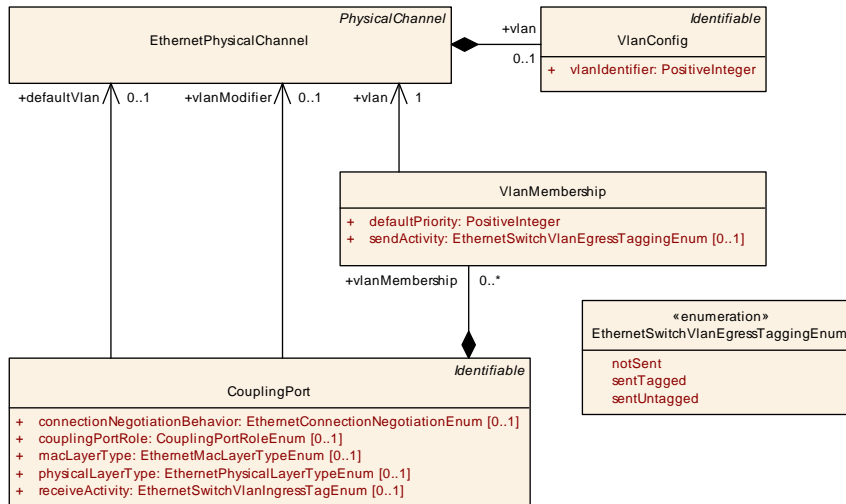


Figure 3.15: VLAN Modification

Enumeration	EthernetSwitchVlanEgressTaggingEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines the VLAN tag sending behavior.
Literal	Description
notSent	will not be sent Tags: atp.EnumerationLiteralIndex=0
sentTagged	sent with its VLAN tag Tags: atp.EnumerationLiteralIndex=1
sentUntagged	sent without a VLAN tag Tags: atp.EnumerationLiteralIndex=2

Table 3.77: EthernetSwitchVlanEgressTaggingEnum

3.3.6.6.5 Semi-static DHCP server configuration

The ECU which manages the Ethernet switch may run a semi-static DHCP server.

[TPS_SYST_03013] **Semi-static DHCP server configuration** [In order to be able to assign always the same IP-address to a dedicated DHCP client, the DHCP server needs the information at which switch port the DHCP request with the specific MAC address has been received. With this switch port information the DHCP server will assign the IP-address according to the `VlanMembership.dhcpAddressAssignment`.

This allows the assignment of MAC addresses by the Tier 1 and assignment of IP addresses by the OEM. With this mechanism it is also possible to assign different IP addresses to several VLANs at the same port. | (RS_SYST_00052)

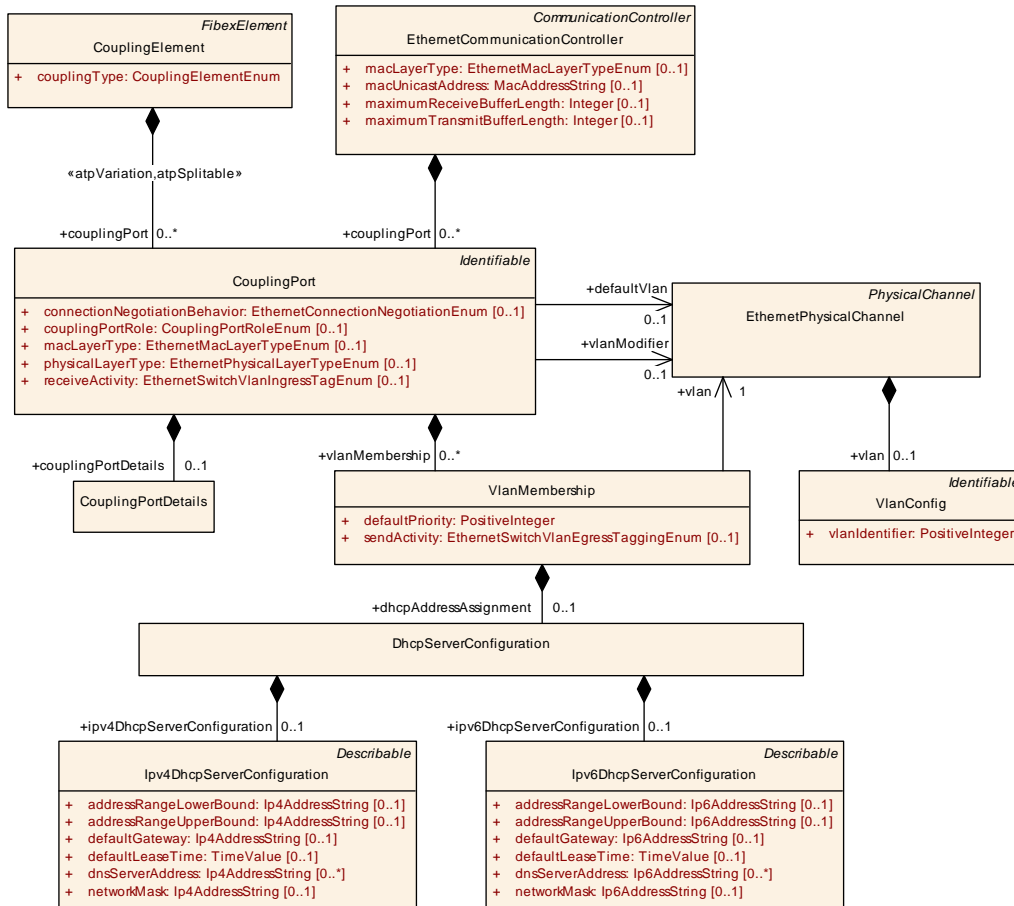


Figure 3.16: Semi-static DHCP configuration

Class	DhcpServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of DHCP servers that are running on the network endpoint. It is possible that an Ipv4DhcpServer and an Ipv6DhcpServer run on the same Ecu.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
ipv4DhcpServer Configuration	Ipv4DhcpServer Configuration	0..1	aggr	Configuration of a IPv4 DHCP server that runs on the network endpoint.
ipv6DhcpServer Configuration	Ipv6DhcpServer Configuration	0..1	aggr	Configuration of a IPv6 DHCP server that runs on the network endpoint.

Table 3.78: DhcpServerConfiguration

Class	Ipv4DhcpServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of a IPv4 DHCP server that runs on the network endpoint.			
Base	<i>ARObject, Describable</i>			
Attribute	Type	Mult.	Kind	Note
addressRange LowerBound	Ip4AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv4 Address. Notation: 255.255.255.255.
addressRange UpperBound	Ip4AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. Pv4 Address. Notation: 255.255.255.255.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLease Time	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation 255.255.255.255 Tags: xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip4AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 3.79: Ipv4DhcpServerConfiguration

Class	Ipv6DhcpServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of a IPv6 DHCP server that runs on the network endpoint.			
Base	<i>ARObject, Describable</i>			
Attribute	Type	Mult.	Kind	Note
addressRange LowerBound	Ip6AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:....FFFF.
addressRange UpperBound	Ip6AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:....FFFF.
defaultGateway	Ip6AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLease Time	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation: FFFF:....FFFF. Tags: xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip6AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 3.80: Ipv6DhcpServerConfiguration

3.3.7 CDD

The System Template allows the integration of custom bus systems on the topology level.

[TPS_SYST_01127] CDD Topology support [The elements `UserDefinedCluster`, `UserDefinedPhysicalChannel`, `UserDefinedCommunicationConnector` and `UserDefinedCommunicationController` can be used to describe alternative communication technologies (e.g. I2C, USB, serial line) that are integrated in AUTOSAR as Complex Drivers.] ([RS_SYST_00044](#))

The Pdu-based communication via Complex Drivers is described in chapter 6.13.

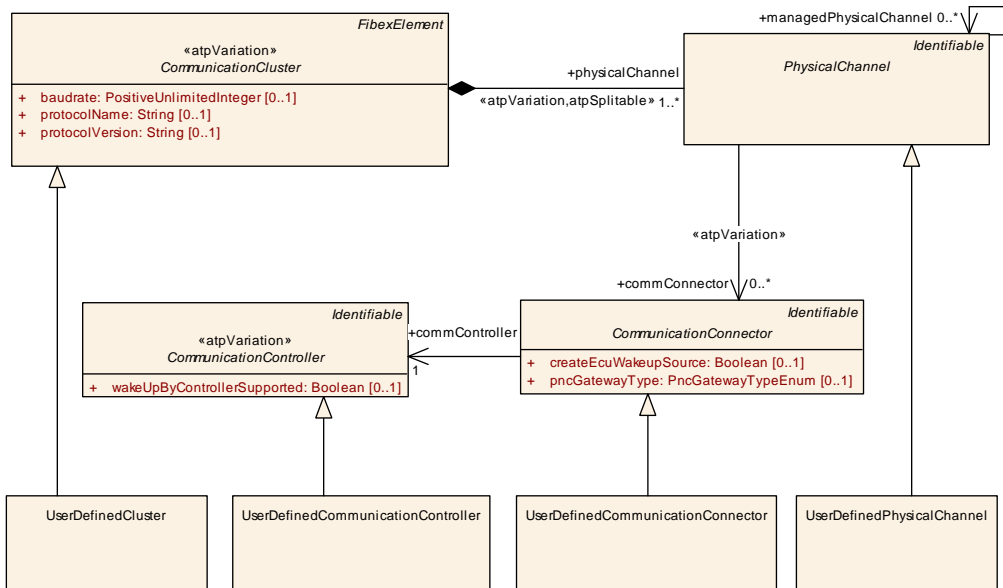


Figure 3.17: User defined topology elements

Class	<<atpVariation>> UserDefinedCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modeling of arbitrary Communication Clusters (e.g. bus systems that are not supported by AUTOSAR). Tags: atp.recommendedPackage=CommunicationClusters			
Base	ARObject, CollectableElement, CommunicationCluster, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.81: UserDefinedCluster

Class	UserDefinedPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modeling of arbitrary Physical Channels.			
Base	ARObject, Identifiable , MultilanguageReferrable , PhysicalChannel , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.82: UserDefinedPhysicalChannel

Class	UserDefinedCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modeling of arbitrary Communication Connectors.			
Base	ARObject, CommunicationConnector , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.83: UserDefinedCommunicationConnector

Class	<<atpVariation>> UserDefinedCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modeling of arbitrary Communication Controllers.			
Base	ARObject, CommunicationController , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 3.84: UserDefinedCommunicationController

3.4 Mapping of Topology Entities onto Hardware Elements

As explained in the previous sections, the System Template contains all classes necessary to describe the physical topology in an AUTOSAR system. Based on this description, the communication matrix can be realized as explained in chapter 6.

[TPS_SYST_01019] Mapping of topology elements to elements of the ECU Resource Template [It is possible to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template.] ([RS_SYST_00006](#))

It can be specified which [HwElement](#) is realizing each given [EcuInstance](#), providing the means for algorithms to map software components onto the systems [EcuInstance](#). By specifying which [hwCommunicationPort](#)³ on a [hwCommunicationController](#)⁴ implements the topology's [CommunicationConnector](#) on a [CommunicationController](#), the hardware-oriented parameters in the Communication-drivers may be derived in ECU configuration phase.

³[HwPinGroup](#) which is of category Communication Port

⁴[HwElement](#) which is of category Communication Controller

Please note that this is a rather specific type of mapping, optionally binding ECU-local topology elements to specific hardware resources. It should not be confused with the System Mapping part of the System Description, where system-wide mapping decisions are described, like e.g. the the mapping of Software Components onto ECUs or the mapping of Data Element Prototypes onto System Signals (for the System Mapping, see chapter 5).

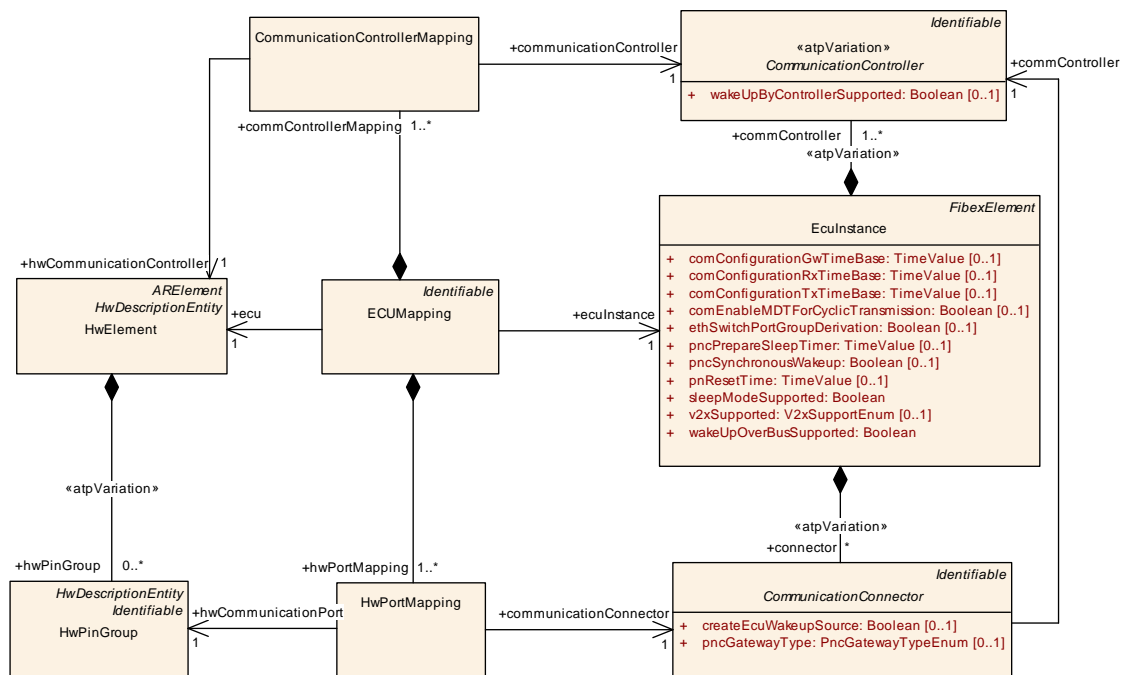


Figure 3.18: Mapping of topology description elements in the System Template onto hardware elements defined in the ECU Resource Template (ECUResourceMapping)

[constr_3006] valid EcuMapping [The referenced `hwCommunicationController` and `hwCommunicationPort` shall be part of the referenced `ecu`.

`ECUMapping.ecu.nestedElement` contains `ECUMapping.commControllerMapping.hwCommunicationController`

`ECUMapping.ecu.nestedElement` contains `ECUMapping.hwPortMapping.hwCommunicationPort`]

3.4.1 ECU Mapping

`ECUMapping` allows to assign a `HwElement` to an `EcuInstance` used in a physical topology.

[TPS_SYST_01013] EcuInstance stands for its own [An `EcuInstance` can be defined in a stand alone and reusable way without a need to have an `ECUMapping`.] (*RS_SYST_00013*)

[constr_3030] valid relationship between ECUMapping and EcuInstance [If an `EcuInstance` is assigned to a `HwElement` the `EcuInstance` shall belong to the same `System` as the `ECUMapping`.]()

[constr_3248] Category of HwElement for ECUMapping [The `HwElement` which is referenced from `ECUMapping` in the role `ecu` shall be of category `MicroController`.]()

There exists an inconsistency between the System Template and the ECU Resource Template concerning the usage of the term "Ecu". In the System Template "Ecu" is used to determine one instance of an AUTOSAR Stack (e.g. like in `EcuInstance`). In the Ecu Resource Template "Ecu" is used to describe the physical box (`HwElement` of category `Ecu`) containing the electronics which may contain several processing units with several AUTOSAR Stack instances running.

Class	ECUMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	ECUMapping allows to assign an ECU hardware type (defined in the ECU Resource Template) to an ECUInstance used in a physical topology.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
commControllerMapping	CommunicationControllerMapping	1..*	aggr	The ECUMapping contains the mapping of all CommunicationControllers of the ECU.
ecu	HwElement	1	ref	Reference to a HwElement of category ECU in the ECU Resource Template.
ecuInstance	EcuInstance	1	ref	Reference to the EcuInstance in the System Template
hwPortMapping	HwPortMapping	1..*	aggr	The ECUMapping contains the mapping of all HW Communication Ports of the ECU.

Table 3.85: ECUMapping

3.4.2 Communication Controller Mapping

[TPS_SYST_01014] Semantics of CommunicationControllerMapping [`CommunicationControllerMapping` specifies the `HwElement` to realize the specified `CommunicationController` in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers.]([RS_SYST_00013](#))

Class	CommunicationControllerMapping
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping
Note	<code>CommunicationControllerMapping</code> specifies the <code>CommunicationPeripheral</code> hardware (defined in the ECU Resource Template) to realize the specified <code>CommunicationController</code> in a physical topology.
Base	<i>ARObject</i>





Class		CommunicationControllerMapping		
Attribute	Type	Mult.	Kind	Note
communication Controller	Communication Controller	1	ref	Reference to the CommunicationController in the System Template
hw Communication Controller	HwElement	1	ref	Reference to a HwElement of category Communication Controller in the ECU Resource Template.

Table 3.86: CommunicationControllerMapping

3.4.3 HW-Port Mapping

[TPS_SYST_01015] Semantics of [HwPortMapping](#) [[HwPortMapping](#) specifies the hardware to realize the specified [CommunicationConnector](#) in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers.] ([RS_SYST_00013](#))

Class		HwPortMapping		
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	HwPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
communication Connector	Communication Connector	1	ref	Reference to the CommunicationConnector in the System Template
hw Communication Port	HwPinGroup	1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the Hw CommunicationController is described in the Ecu Resource Description.

Table 3.87: HwPortMapping

4 Top-level Software Composition

One of the most important inputs for the System Generator is the knowledge about the Application Software Components, their communication capabilities and the connections between them: Each `SystemSignal` (chapter 6.2) that is going to be exchanged between mapped Software Components onto different ECUs is a consequence of a connection between such application Software Components.

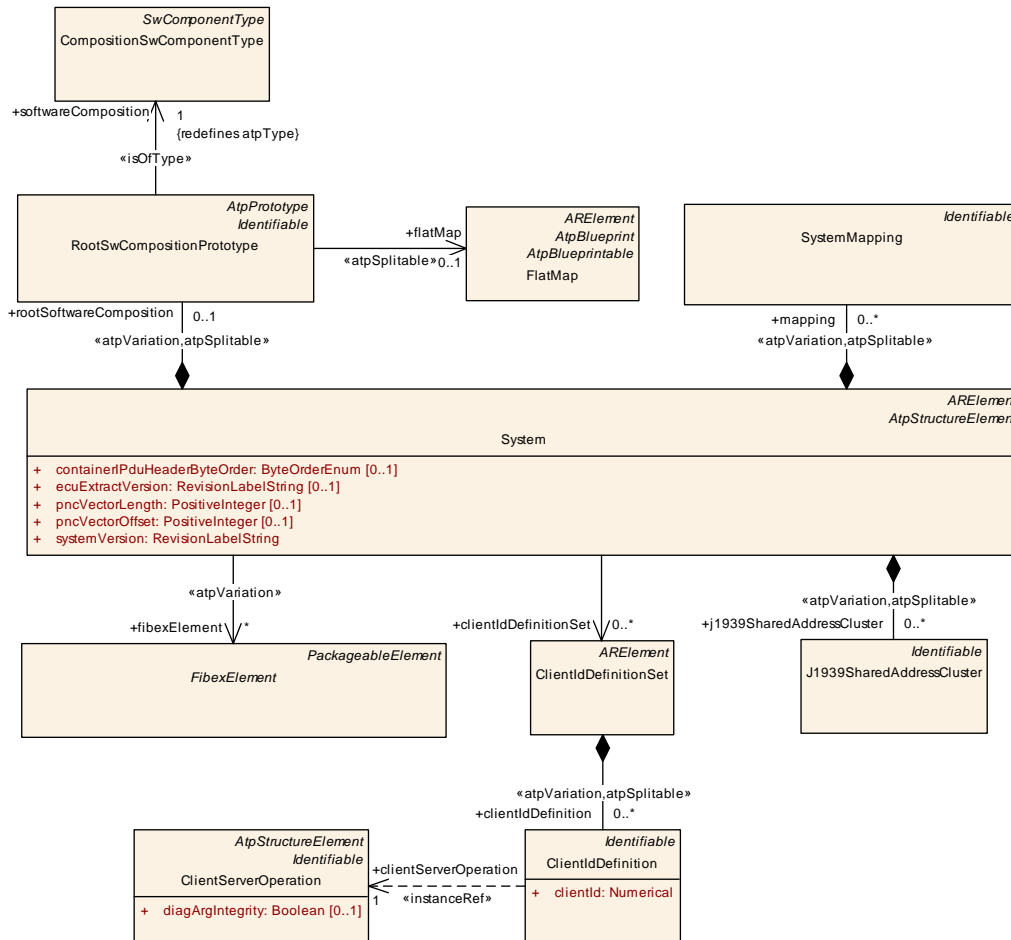


Figure 4.1: Inclusion of a (top-level) Software Composition into an AUTOSAR system (SystemTemplate)

In AUTOSAR, Software Components can either be atomic (`AtomicSwComponentType`) or may consist of a composition of other Software Components `CompositionSwComponentType` [5]. In order to assemble non-trivial applications from AUTOSAR components, such compositions can be built up hierarchically, until the outermost `CompositionSwComponentType` forms a kind of top-level composition.

[constr_3031] Complete System Description does not have ports on the outermost composition [In a complete `System` with `category` `ABSTRACT_SYSTEM_DESCRIPTION` or `System` with `category` `SYSTEM_DESCRIPTION` this outermost `CompositionSwComponentType` has the unique feature that it doesn't have any outside ports, but all the SWC contained in it are connected to each

other and fully specified by their [SwComponentTypes](#), [PortPrototypes](#), [PortInterfaces](#), [VariableDataPrototypes](#), [InternalBehavior](#) etc.]()

[TPS_SYST_01016] System Extract, Ecu System Description and Ecu Extract may have ports [In a [System](#) with [category](#) SYSTEM_EXTRACT and a [System](#) with [category](#) ECU_SYSTEM_DESCRIPTION and a [System](#) with [category](#) ECU_EXTRACT outside ports for the outermost composition are allowed.]([RS_SYST_00027](#))

Since the System/Ecu Extract represents the view on one Ecu, there may be the need to define the communication of this extract with the outside world.

Two approaches are available how the external communication of an ECU in the System Extract is described. In section 11.2 the communication mapping is performed in the hierarchical structure of software components. In section 11.3 external communication delegation ports are added to the System extract outermost composition. Each delegated port is connected via a [DelegationSwConnector](#) with ports of the included components that are used for the external communication.

A [System](#) considers such a top-level [CompositionSwComponentType](#) as its application software system input by owning exactly one [RootSwCompositionPrototype](#) class, which points to the [CompositionSwComponentType](#) forming the input via its `«isOfType»` relationship as shown in Figure 4.1.

[TPS_SYST_01017] The role of the top-level software composition [An AUTOSAR [System](#) uses the specialized prototype class [RootSwCompositionPrototype](#) in order to designate the referenced [CompositionSwComponentType](#) as the top-level software composition.]([RS_SYST_00006](#))

Class	RootSwCompositionPrototype			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	<p>The RootSwCompositionPrototype represents the top-level-composition of software components within a given System. According to the use case of the System, this may for example be the a more or less complete VFB description, the software of a System Extract or the software of a flat ECU Extract with only atomic SWCs.</p> <p>Therefore the RootSwComposition will only occasionally contain all atomic software components that are used in a complete VFB System. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System. The internal structure of such a component contains often substantial intellectual property of a supplier. Therefore a top-level software composition will often contain empty compositions which represent subsystems.</p> <p>The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.</p>			
Base	ARObject , AtpFeature , AtpPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
calibrationParameterValueSet	CalibrationParameterValueSet	*	ref	Used CalibrationParameterValueSet for instance specific initialization of calibration parameters. Stereotypes: atpSplittable Tags: atp.Splitkey=calibrationParameterValueSet





Class	RootSwCompositionPrototype			
flatMap	FlatMap	0..1	ref	The FlatMap used in the scope of this RootSw CompositionPrototype. Stereotypes: atpSplitable Tags: atp.Splitkey=flatMap
software Composition	CompositionSw ComponentType	1	tref	We assume that there is exactly one top-level composition that includes all Component instances of the system Stereotypes: isOfType

Table 4.1: RootSwCompositionPrototype

5 Mapping

A central part of the system generation process is the mapping of software components (*SwComponentPrototypes*) to ECUs, and the subsequent mapping of the communication between these software components to bus frames. Input to the software component mapping is the *RootSwCompositionPrototype*, which describes which software components have to be mapped, and the System Topology, which defines the ECU instances that are available as mapping targets. Once this mapping is done, also the communication matrix has to be taken into account for the next mapping step, the mapping of data elements exchanged between software components to bus frames. This communication matrix may either be predefined, or may be generated as part of this second mapping step. In the metamodel, different aspects of these mapping are aggregated by the meta class *SystemMapping*, as shown in Figure 5.1.

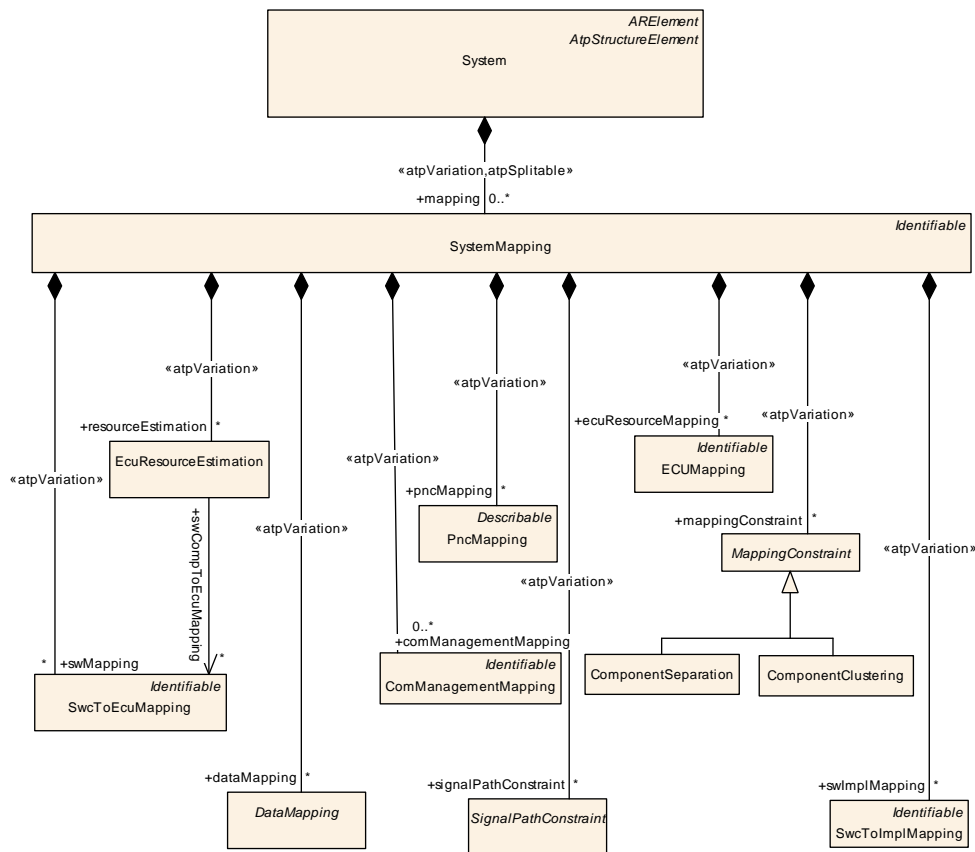


Figure 5.1: Mapping Overview (Mapping)

The following mappings are defined:

- The `SwcToEcuMapping` meta-class maps one or several `SwComponentPrototypes` to ECUs. In the System Constraint Description it is possible to predefine the mapping of `SwComponentPrototypes` to ECUs. The predefinition limits the system architect's freedom to map software components to arbitrary ECUs. After the system generation in the System Configuration Description, all atomic software components that are directly or indirectly part of the top level composition shall be mapped with this mapping rule. Software component mapping is described in detail in chapter 5.1.
- The meta-class `EcuResourceEstimation` specifies the resource estimation for RTE and basic software (see chapter 5.3).
- The `ECUMapping` meta-class is used to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template (see chapter 3.4).
- The `DataMapping` meta-class is used to map `VariableDataPrototypes` and `ClientServerOperations` in software component ports (i.e. the data exchanges between software components) to signals. The data mapping is described in detail in chapter 5.2.
- The `ComManagementMapping` defines the mapping of one or several Mode Management `PortGroups` and communication channels (see chapter 5.5).
- The `PncMapping` defines the Partial Network behavior (see chapter 5.4).
- The `SignalPathConstraint` meta-class is used to define which specific way a signal (data element or client server operation arguments) between two Software Components should take in the network without defining in which frame and with which timing it is transmitted. This Signal Path Constraint is introduced in chapter 5.2.2.
- The `MappingConstraint` meta-class is used to define constraints that constrain the mapping of software components. It's sub-classes allow to constraint which `SwComponentPrototypes` shall be mapped together on the same ECU (`ComponentClustering`) and which shall not be mapped to the same ECU (`ComponentSeparation`). The mapping constraints are described in detail in chapter 5.1.4.
- The `J1939ControllerApplicationToJ1939NmNodeMapping` maps a Software Component to which a standardized function id is assigned to a `J1939NmNode` (see chapter 5.1.5)
- Finally, the `SwcToImplMapping` meta-class is used to assign one `Implementation` to one or more `SwComponentPrototypes` (see chapter 5.1.2).

Class	SystemMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The system mapping aggregates all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
application PartitionToEcu Partition Mapping	ApplicationPartitionToEcuPartitionMapping	*	aggr	Mapping of ApplicationPartitions to EcuPartitions Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
com Management Mapping	ComManagementMapping	*	aggr	Mappings between Mode Management PortGroups and communication channels. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime
cryptoService Mapping	CryptoServiceMapping	*	aggr	This aggregation represents the collection of crypto service mappings in the context of the enclosing System Mapping. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
dataMapping	DataMapping	*	aggr	The data mappings defined. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
ecuResource Mapping	ECUMapping	*	aggr	Mapping of hardware related topology elements onto their counterpart definitions in the ECU Resource Template. atpVariation: The ECU Resource type might be variable. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime
j1939Controller ApplicationTo J1939NmNode Mapping	J1939ControllerApplicationToJ1939NmNodeMapping	*	aggr	Mapping of a J1939ControllerApplication to a J1939Nm Node.
mapping Constraint	MappingConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of SW components to ECUs. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime
pncMapping	PncMapping	*	aggr	Mappings between Virtual Function Clusters and Partial Network Clusters. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime
resource Estimation	EcuResourceEstimation	*	aggr	Resource estimations for this set of mappings, zero or one per ECU instance. atpVariation: Used ECUs are variable. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime





Class	SystemMapping			
signalPathConstraint	SignalPathConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of data elements to signals. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime
swcToApplicationPartitionMapping	SwcToApplicationPartitionMapping	*	aggr	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
swImplMapping	SwcToImplMapping	*	aggr	The mappings of AtomicSoftwareComponent Instances to Implementations. atpVariation: Derived, because SwcToEcuMapping is variable. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
swMapping	SwcToEcuMapping	*	aggr	The mappings of SW components to ECUs. atpVariation: SWC shall be mapped to other ECUs. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table 5.1: SystemMapping

5.1 Software Component Mapping

A fundamental concept of AUTOSAR is that SW components may be developed independently of a specific ECU hardware, and can be mapped to an ECU in the AUTOSAR System Generation Process. The System Constraint Description acts as an input to this System Generation Phase. Nevertheless, there may be some SW components which are already mapped due to previous iterations of the system generation step, and there may be system constraints that limit the system architect's freedom to map SW components to arbitrary ECUs. In the following, the individual elements are described in more detail.

5.1.1 SW Component to ECU Mapping

[TPS_SYST_01001] Definition of [SwcToEcuMapping](#) [With the [SwcToEcuMapping](#) element it is possible to express the mapping of [SwComponentPrototypes](#) to one [EcuInstance](#) or optional to individual [HwElements](#) with [category](#) Processing Unit residing in this ECU. An optional assignment of Sensor/Actuator [SwComponentPrototypes](#) to Sensor/Actuator [HwElements](#) is also possible.]([RS_SYST_00007](#), [RS_SYST_00033](#))

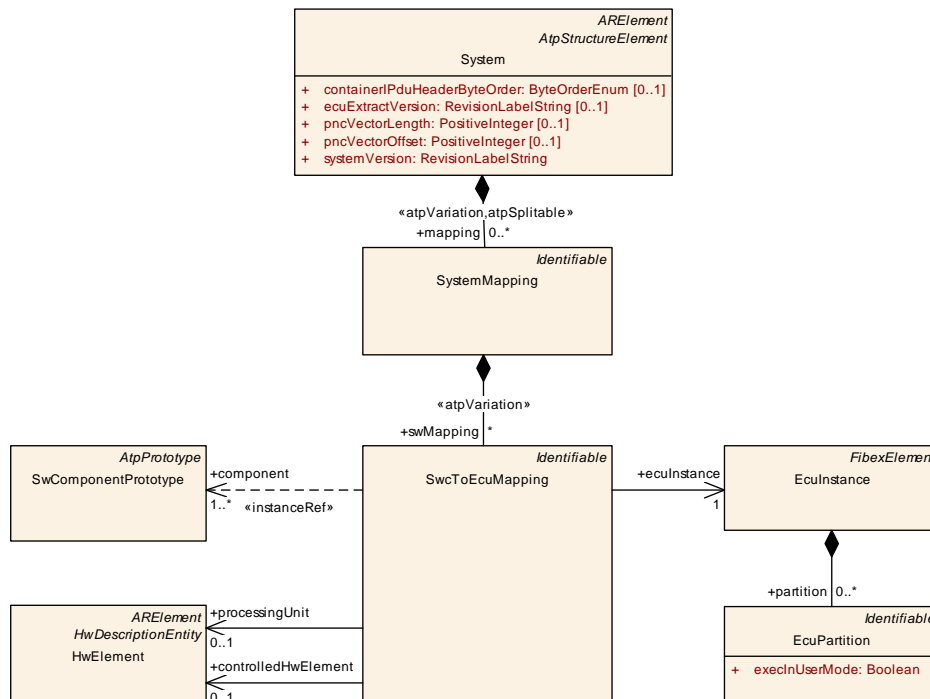


Figure 5.2: SW component to ECU mapping (SwcToEcuMapping)

The `SwcToEcuMapping` collects a list of all `SwComponentPrototypes` that shall be deployed onto the associated `SwcToEcuMapping` targets.

[TPS_SYST_02114] Mapping of `SwComponentPrototypes` onto `SwcToEcuMapping` targets [The `SwcToEcuMapping` of `SwComponentPrototypes` to

- `EcuInstance`
- `processingUnit`
- `controlledHwElement`

is arbitrary.

It is equivalent to either

- have several `SwcToEcuMappings` which map a set of `SwcToEcuMapping.components` to a `SwcToEcuMapping.ecuInstance`, `SwcToEcuMapping.processingUnit`, `SwcToEcuMapping.controlledHwElement`,
- or one `SwcToEcuMapping` which maps the set of `SwcToEcuMapping.components` at once.

]([RS_SYST_00007](#))

[constr_3263] Restriction of usage of `SwcToEcuMapping` in a `System` [For all `SwcToEcuMappings` in a `System` the following restriction applies: No two `SwcToEcuMappings` shall have the exact same reference to

- `SwComponentPrototype`

- `EcuInstance`
- `processingUnit`
- `controlledHwElement`

]()

`SwcToEcuMapping` may map either prototypes of `AtomicSwComponentType` or those of `CompositionSwComponentType`.

[TPS_SYST_01020] Unconditional mapping of atomic Software Components [In case a prototype of an atomic Software Components is mapped, the mapping is unconditional.](*RS_SYST_00007*)

[TPS_SYST_01021] Mapping of `CompositionSwComponentType` [In case a mapped `SwComponentPrototype` refers to a `CompositionSwComponentType`, the mapping is applied to any inner `SwComponentPrototype` recursively; however, it may be overwritten by additional `SwcToEcuMapping` mapping inner `SwComponentPrototype` to different `EcuInstances`.](*RS_SYST_00007*)

Usually a particular component prototype can be mapped explicitly to at most one ECU in a given system (leaving aside variant handling and the implicit mapping of "inner" prototypes mentioned above) but there are two exceptions:

- **[TPS_SYST_01022] Prototype of a `ParameterSwComponentType` can be mapped to more than one ECU** [A prototype of a `ParameterSwComponentType` can be mapped to more than one ECU. This is required, because this special component does not communicate over the network, so that a copy of the prototype has to be created on each ECU where it is required.](*RS_SYST_00007*)
- **[TPS_SYST_01023] Prototype of an `ServiceProxySwComponentType` can be mapped to more than one ECU** [A prototype of an `ServiceProxySwComponentType` can be mapped to more than one ECU even if it appears only once in the VFB system, because a prototype of this special component is required on each ECU, for which local Services are addressed via the proxy.](*RS_SYST_00031*)

[constr_3021] Mapping of `SensorActuatorSwComponents` to `SensorActuatorHwElements` [Only `SwComponentPrototypes` that are typed by `SensorActuatorSwComponentType` shall be mapped to a `HwElement` with `category` `SensorActuator` via the `controlledHwElement` relation.]()

[constr_3249] Category of `HwElement` for `SwcToEcuMapping` [The `HwElement` which is referenced from `SwcToEcuMapping` in the role `processingUnit` shall be of category "ProcessingUnit".]()

The following table describes the `SwcToEcuMapping` in detail.

Class	SwcToEcuMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Map software components to a specific ECU Instance and optionally to a processing unit and to an Ecu Partition. For each combination of ECUInstance and the optional ProcessingUnit and the optional Ecu Partition and the optional SensorActuator only one SwcToEcuMapping shall be used.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
component	SwComponent Prototype	1..*	iref	References to the software component instances that are mapped to the referenced ECUInstance. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ECU. If there is additionally a mapping of some SwComponent Prototype INSIDE the Composition to another ECU Instance the inner mapping overrides the outer mapping.
controlledHwElement	HwElement	0..1	ref	Optional mapping of SwComponentPrototypes that are typed by SensorActuatorSwComponentType to a HwElement with category SensorActuator.
ecuInstance	EcuInstance	1	ref	Reference to a specific ECU Instance description.
processingUnit	HwElement	0..1	ref	Optional mapping of software components to individual microcontroller cores residing in one ECU. A microcontroller core is described in the ECU Resource Template by the HwElement of HwCategory Processing Unit.

Table 5.2: SwcToEcuMapping

5.1.2 Software Component to Implementation Mapping

As several implementations may exist for the same `AtomicSwComponentType`, it needs to be decided on and specified which instances of a given `AtomicSwComponentType` are mapped to which `Implementation`. According to the AUTOSAR Methodology this information can either be added within the `Configure System` activity, or later when the RTE part is configured during `Configure ECU` phase. If the mapping is done in System Configuration, a `SwcToImplMapping` is being used for assigning one `Implementation` to one or more instances of `SwComponentPrototype` relating to the same `AtomicSwComponentType`. This is illustrated in Figure 5.3.

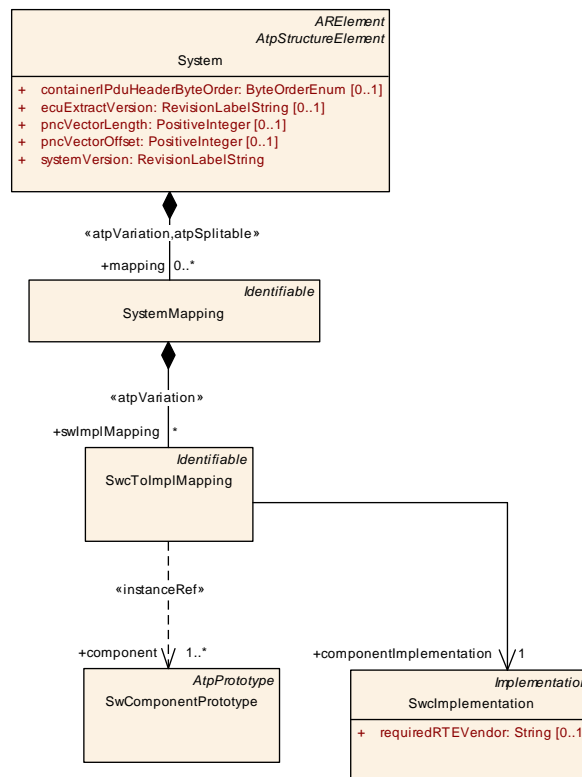


Figure 5.3: SW Component to Implementation mapping (SwcToImplMapping)

[constr_3002] valid swcToImplMapping [The referenced `SwcImplementation` refers to a `SwcInternalBehavior` that is part of a `AtomicSwComponentType`. The same `AtomicSwComponentType` shall be the type of the referenced `SwComponentPrototype`.

```
SwcToImplMapping.componentImplementation.behavior.component == SwcToImplMapping.component.type] ()
```

The following table contains the detailed description of [SwcToImplMapping](#):

Class	SwcToImplMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Map instances of an AtomicSwComponentType to a specific Implementation.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
component	SwComponentPrototype	1..*	iref	Reference to the software component instances that are being mapped to the specified Implementation. The targeted SwComponentPrototype needs be of the AtomicSwComponentType being implemented by the referenced Implementation.
component Implementation	SwImplementation	1	ref	Reference to a specific Implementation description. Implementation to be used by the specified SW component instance. This allows to achieve more precise estimates for the resource consumption that results from mapping the instance of an atomic SW component onto an ECU.

Table 5.3: SwcToImplMapping

5.1.3 SW Component to Partition Mapping

With the [SwcToApplicationPartitionMapping](#) and the [ApplicationPartitionToEcuPartitionMapping](#) an OEM has the option to predefine an allocation to memory partitions in the System Design phase. The final and complete assignment is described in the OS Configuration. The [SwcToApplicationPartitionMapping](#) defines a mapping to [ApplicationPartitions](#) that allows an allocation to a formally defined partition at a point in time when the [EcuInstance](#) is not yet known or defined. In a later methodology step this assignment can be refined with the [ApplicationPartitionToEcuPartitionMapping](#) to an [EcuPartition](#) defined in the context of an [EcuInstance](#).

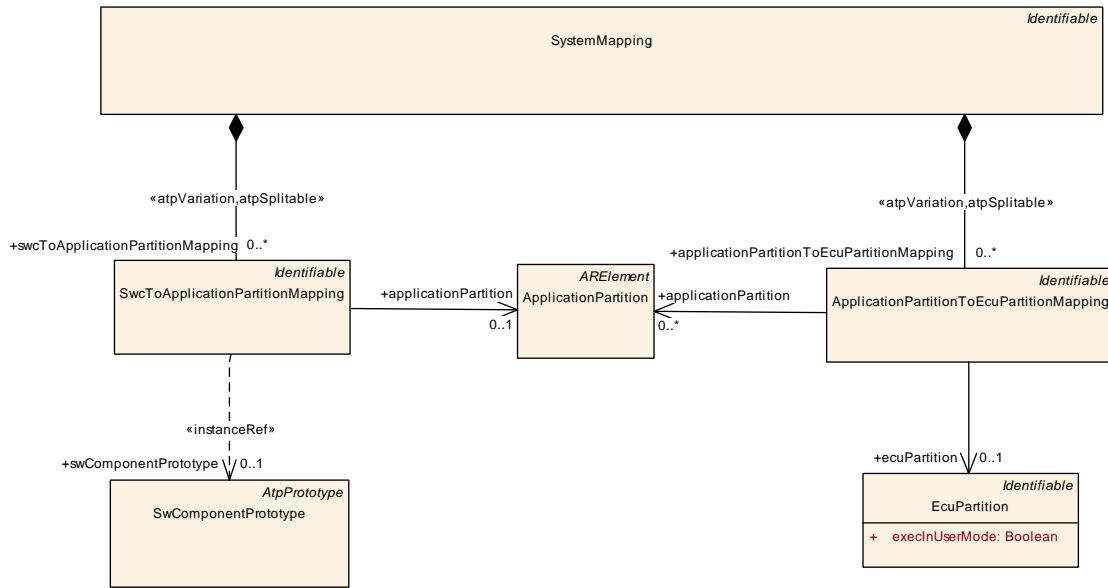


Figure 5.4: SW Component to Application Partition mapping

Class	SwcToApplicationPartitionMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
application Partition	ApplicationPartition	0..1	ref	Reference to an ApplicationPartition to which a Sw ComponentPrototype is mapped.
swComponent Prototype	SwComponent Prototype	0..1	iref	References to the software component instances that are mapped to the referenced ApplicationPartition. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ApplicationPartition. If there is additionally a mapping of some SwComponent Prototype INSIDE the Composition to another Application Partition the inner mapping overrides the outer mapping.

Table 5.4: SwcToApplicationPartitionMapping

Class	ApplicationPartition			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	ApplicationPartition to which SwComponentPrototypes are mapped at a point in time when the corresponding EcuInstance is not yet known or defined. In a later methodology step the Application Partition can be assigned to an EcuPartition. Tags: atp.recommendedPackage=ApplicationPartitions			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	Type	Mult.	Kind	Note





Class	ApplicationPartition			
-	-	-	-	-

Table 5.5: ApplicationPartition

Class	ApplicationPartitionToEcuPartitionMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Maps ApplicationPartitions to EcuPartitions. With this mapping an OEM has the option to predefine an allocation of Software Components to EcuPartitions in the System Design phase. The final and complete assignment is described in the OS Configuration.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
application Partition	ApplicationPartition	*	ref	Reference to ApplicationPartitions that are mapped to an EcuPartition.
ecuPartition	EcuPartition	0..1	ref	Reference to EcuPartition to which the Application Partitions are assigned.

Table 5.6: ApplicationPartitionToEcuPartitionMapping

Class	EcuPartition			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Partitions are used as error containment regions. They permit the grouping of SWCs and resources and allow to describe recovery policies individually for each partition. Partitions can be terminated or restarted during run-time as a result of a detected error.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
execInUser Mode	Boolean	1	attr	A partition can execute either in CPU user mode (execInUserMode = TRUE) or supervisor mode (execInUserMode = FALSE). In user mode, the partition has a limited access to memory, to memory mapped hardware and to CPU. In user mode, the partition is mapped to a non-trusted OS-Application.

Table 5.7: EcuPartition

[constr_3232] [ApplicationPartition](#) is allowed to be mapped to only one [EcuPartition](#) [Each [ApplicationPartition](#) shall be mapped at most once to an [EcuPartition](#) via the [ApplicationPartitionToEcuPartitionMapping](#).]()

[constr_3229] [SwComponentPrototype](#) mapped to an [ApplicationPartition](#) and [EcuInstance](#) [If the [SwcToEcuMapping.ecuInstance](#) exists then a [SwComponentPrototype](#) that is mapped to an [ApplicationPartition](#) via the [SwcToApplicationPartitionMapping](#) shall only be mapped by an [ApplicationPartitionToEcuPartitionMapping](#) to an [EcuPartition](#) that is aggregated by the [EcuInstance](#) referenced by means of [SwcToEcuMapping.ecuInstance](#).]()

5.1.4 Software Component Mapping Constraints

In contrast to the mapping description described in the previous chapters, mapping constraints allow to define invariants that have to be fulfilled by a valid mapping. They are aggregated in the `MappingConstraint` element as introduced in chapter 5 and depicted Figure 5.1. This chapter describes which mapping constraints can be described in the System Constraint Description. The description of this meta-class can be found in the following table:

Class	<i>MappingConstraint</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Different constraints that may be used to limit the mapping of SW components to applicable ECUs, Partitions or Cores depending on the mappingScope attribute.			
Base	<i>ARObject</i>			
Subclasses	<code>ComponentClustering</code> , <code>ComponentSeparation</code>			
Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the mapping constraint.

Table 5.8: MappingConstraint

The two constraints (`ComponentClustering` and `ComponentSeparation`) shown in Figure 5.5 express the restrictions that Software Components impose on each other when performing the mapping onto the ECUs, Cores or Partitions. In fact, before the mapping process begins, it can be useful to impose the allocation of a predefined set of SW components onto the same ECU, especially if such a set is tightly linked from a functional point of view. In the same way, two critical SW components, performing some kind of redundancy, may be not suitable to run both on the same ECU. Thus, we call these two kinds of mapping constraints, respectively, `ComponentClustering` and `ComponentSeparation`.

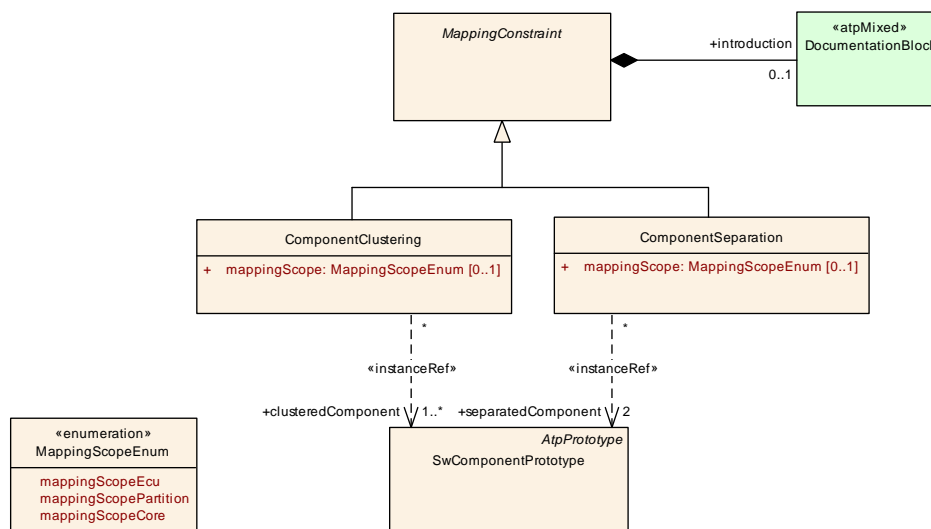


Figure 5.5: Details on ComponentClustering and ComponentSeparation (SwcClustering)

5.1.4.1 ComponentClustering

[TPS_SYST_01024] Component Clustering [The [ComponentClustering](#) constraint (also, *clustering*) is to be used for expressing that a certain set of SW components (atomic or not) shall be mapped (allocated) onto the same ECU, Core, Partition depending on the defined [mappingScope](#) attribute.] ([RS_SYST_00008](#))

This is some kind of "execute together on same ECU" constraint.

The semantic of the clustering constraint is straightforward if all referenced SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01025] Clustering of Compositions [All of the atomic SW components making up the composition shall be mapped onto the same ECU, Core, Partition depending on the defined [mappingScope](#) attribute together with all other SW components (atomic or not) referenced by the constraint.] ([RS_SYST_00008](#))

This also means that a *clustering* constraint can also refer to only a single composition.

A *clustering* constraint is part of a [MappingConstraint](#) element and it shall refer to one or more [SwComponentPrototype](#) elements, representing the instances of the SW component(s) that shall be mapped together.

Class	ComponentClustering			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Constraint that forces the mapping of all referenced SW component instances to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mappingScope is not specified then mappingScopeEcu shall be assumed.			
Base	ARObject , MappingConstraint			
Attribute	Type	Mult.	Kind	Note
clustered Component	SwComponentPrototype	1..*	iref	Reference to the components that have to be mapped together.
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the ComponentClustering mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScopeEcu shall be assumed. Tags: atp.Status=shallBecomeMandatory

Table 5.9: ComponentClustering

Enumeration	MappingScopeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping
Note	Defines the scope for the mapping constraints.
Literal	Description
mappingScopeCore	The mapping constraint applies to different Cores. Tags: atp.EnumerationLiteralIndex=0
mappingScopeEcu	The mapping constraint applies to different Ecus. Tags: atp.EnumerationLiteralIndex=1





Enumeration	MappingScopeEnum
mappingScope Partition	The mapping constraint applies to different Partitions. Tags:atp.EnumerationLiteralIndex=2

Table 5.10: MappingScopeEnum

5.1.4.2 ComponentSeparation

[TPS_SYST_01045] Component Separation [The [ComponentSeparation](#) constraint (also, *separation*) is to be used for expressing that two SW components (atomic or not) shall not be mapped (allocated) onto the same ECU, Core, Partition depending on the defined [mappingScope](#) attribute.] ([RS_SYST_00009](#))

This is some kind of “do not execute together on same ECU” constraint.

The semantic of the separation constraint is straightforward if one or both SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01026] Separation of Compositions [Any of the atomic SW components making up the first composition, shall not be mapped onto the same ECU, Core, Partition depending on the defined [mappingScope](#) attribute with any atomic SW component from the second composition.] ([RS_SYST_00009](#))

As a consequence, and to preserve consistency, an atomic SW component instance cannot be part of two compositions concerned by the same separation constraint, i.e. the two compositions have to be disjoint with regards to component instances¹.

A *separation* constraint is part of a [MappingConstraint](#) element and it shall refer to two [SwComponentPrototype](#) elements, representing the two SW component instances that shall not be allocated together.

Class	ComponentSeparation
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping
Note	Constraint that forces the two referenced SW components (called A and B in the following) not to be mapped to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mapping Scope is not specified then mappingScopeEcu shall be assumed. If a SW component (e.g. A) is a composition, none of the atomic SW components making up the A composition shall be mapped together with any of the atomic SW components making up the B composition. Furthermore, A and B shall be disjoint.



¹The only case where a component instance could be in both sets is if the [ComponentSeparation](#) refers to two elements where one of them is a substructure of the other. Consider the case that Atomic SW Component A is aggregated by composition B, which in turn is aggregated by composition C. Then instance A is both in B and C. It is not a good idea to formulate a separation constraint stating that B and C should not be on the same ECU.



Class	ComponentSeparation			
Base	<i>ARObject</i> , <i>MappingConstraint</i>			
Attribute	Type	Mult.	Kind	Note
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the Component Separation mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScopeEcu shall be assumed. Tags: atp.Status=shallBecomeMandatory
separated Component	SwComponent Prototype	2	iref	The two components that have to be mapped to different ECUs

Table 5.11: ComponentSeparation

[constr_3004] Clustering and separation shall be exclusive [Clustering and separation shall be exclusive, i.e. it SHALL NOT be possible that two [SwComponentPrototypes](#) A and B are associated by a [ComponentClustering](#) and by a [ComponentSeparation](#).]()

5.1.5 J1939 Controller Application Mapping

J1939 is not restricted to mere communication protocols. It also specifies the communication of software functions (a.k.a. J1939 Controller Applications) and thus has a very dedicated view on the software of an automotive ECU. The approach taken by J1939 with respect to software is very similar to the way AUTOSAR specifies software-components.

However, J1939 uses a different terminology and associates such a software-component with a predefined function. In addition, every function in J1939 has a standardized id. This function id is distributed by the Controller Application to the network as part of the so-called "name" which is a unique identifier representing a Controller Application within the J1939 network management.

Controller Applications, to some extent, fulfill the role of a "virtual ECU" since they are visible as independent entities on a J1939 network. In terms of AUTOSAR modeling, the role of a "virtual ECU" for J1939 Controller Applications is fulfilled by the meta-class [J1939NmNode](#).

In order to make use of the AUTOSAR modeling approach for J1939 it is very helpful to associate a standardized function id with a software-component during an early phase of a development project. This function id shall later be mapped to a [J1939NmNode](#) with the identical `J1939NmNode.nodeName.function`.

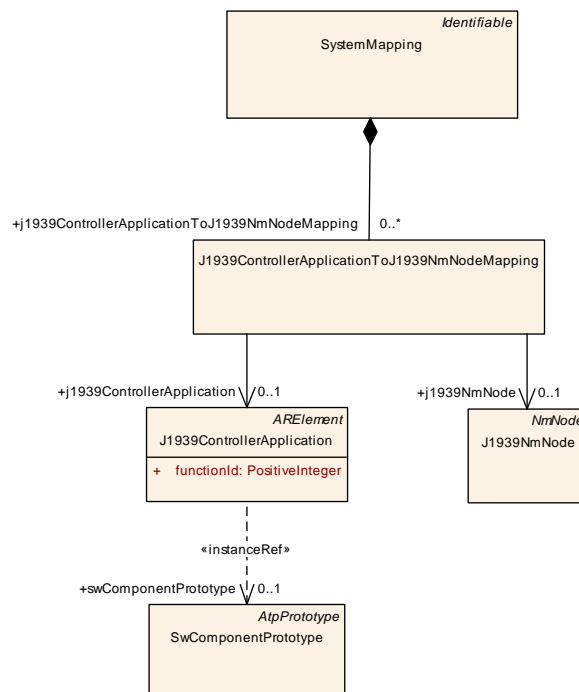


Figure 5.6: J1939 Controller Application to J1939NmNode Mapping

Class	J1939ControllerApplicationToJ1939NmNodeMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	This meta-class represents the ability to map a J1939ControllerApplication to a J1939NmNode. Note that this is similar but not identical to the mapping of SwComponentPrototypes to EcuInstances; for J1939 the semantics of an EcuInstance itself is basically replaced by a J1939NmNode.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
j1939ControllerApplication	J1939ControllerApplication	0..1	ref	Reference to the J1939 Controller Application that is mapped to the referenced J1939NmNode.
j1939NmNode	J1939NmNode	0..1	ref	J1939NmNode that is the target of the J1939ControllerApplicationToJ1939NmNodeMapping.

Table 5.12: J1939ControllerApplicationToJ1939NmNodeMapping

Class	J1939ControllerApplication			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	This element represents a J1939 controller application. Tags: atp.recommendedPackage=J1939ControllerApplications			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
functionId	PositiveInteger	1	attr	This attribute represents the numerical function id of the J1939 controller application.
swComponentPrototype	SwComponentPrototype	0..1	iref	This represents the SwComponentPrototype (which is typically typed by a CompositionSwComponentType) that corresponds to the J1939ControllerApplication.

Table 5.13: J1939ControllerApplication

[constr_3239] Consistent mapping of software-component to J1939NmNode [The value of attribute `J1939NmNode.nodeName.function` of a `J1939NmNode` referenced by `J1939ControllerApplicationToJ1939NmNodeMapping` in the role `j1939NmNode` shall be identical to the value of `J1939ControllerApplication.functionId`.]()

[constr_3240] Consistent mapping of J1939ControllerApplication to EcuInstance [A `SwComponentPrototype` that is referenced by a `J1939ControllerApplication` mapped to a specific `J1939NmNode` shall only be mapped to an `EcuInstance` that in turn owns the same `J1939NmNode`.]()

5.2 Data Mapping

The data mapping description may either be mapping of client server communication or sender receiver communication (see Figure 5.7). It is used to map `VariableDataPrototypes` or `ClientServerOperations` of SW Component Ports to `SystemSignals`.

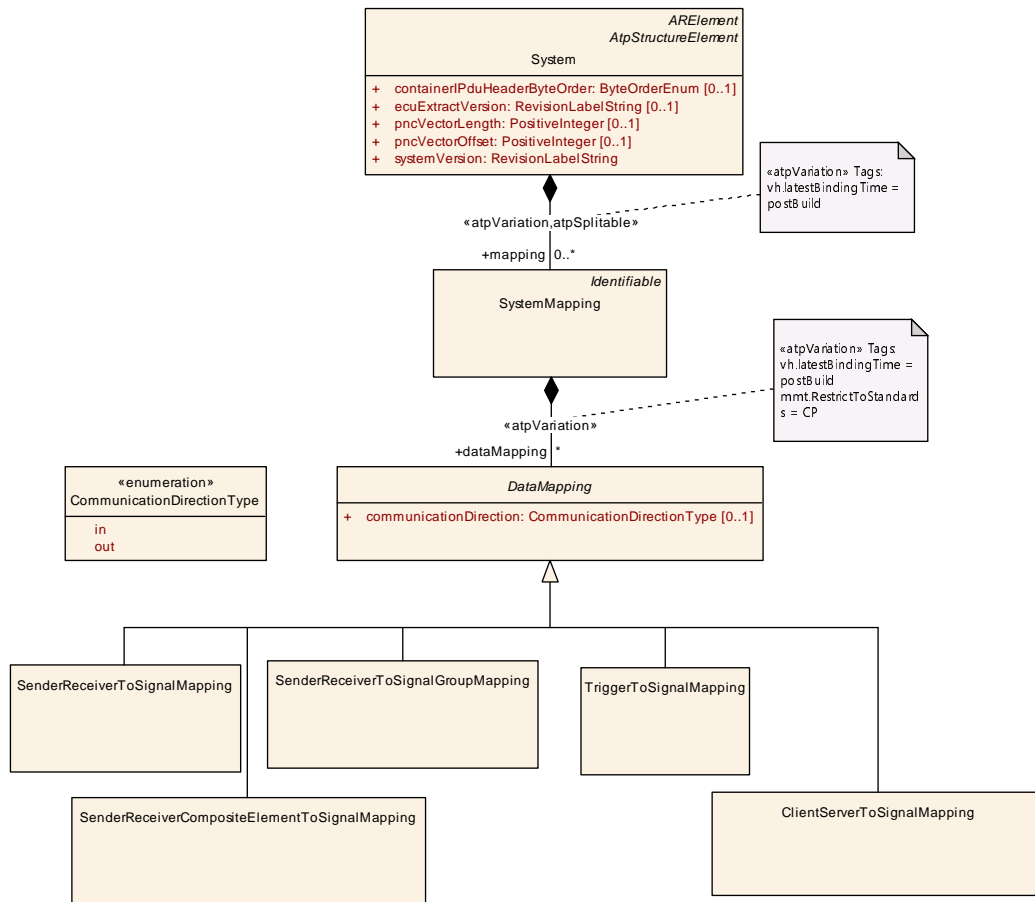


Figure 5.7: Overview: Data Mapping Description (DataMappingOverview)

[TPS_SYST_01030] Representation of **VariableDataPrototypes** and **ClientServerOperations** in System Description [SystemSignals represent VariableDataPrototypes and ClientServerOperations in the communication description.] (RS_SYST_00025)

[constr_5055] **DataMapping** of elements of **PRPortPrototypes** is not supported [A DataMapping shall not map elements of PRPortPrototypes to SystemSignals]()

In other words the usage of **PRPortPrototypes** for inter-Ecu communication is not supported by AUTOSAR.

[TPS_SYST_01032] Independence of **SystemSignals** from **Communication-Clusters** [The SystemSignals can be defined independently of Communication-Clusters.]()

This chapter describes how the **VariableDataPrototypes** and **ClientServer-Operations** are mapped onto **SystemSignals**. The Communication chapter (6) describes how the **SystemSignals** are mapped into Pdus and Frames, implementing the actual inter-ECU communication.

Class	<i>DataMapping</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	<i>ARObject</i>			
Subclasses	ClientServerToSignalMapping , SenderReceiverCompositeElementToSignalMapping , SenderReceiverToSignalGroupMapping , SenderReceiverToSignalMapping , TriggerToSignalMapping			
Attribute	Type	Mult.	Kind	Note
communication Direction	CommunicationDirectionType	0..1	attr	This attribute controls the direction into which the mapped SystemSignal is communicated with respect to the kind of PortPrototype used as the context element of the Data Mapping.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping.

Table 5.14: DataMapping

Class	SystemSignal			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances. Tags: atp.recommendedPackage=SystemSignals			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
dynamicLength	Boolean	1	attr	The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).
physicalProps	SwDataDefProps	0..1	aggr	Specification of the physical representation.

Table 5.15: SystemSignal

A [SystemSignal](#) is used to represent [VariableDataPrototypes](#) for network transport.

[TPS_SYST_01144] Physical properties of a [SystemSignal](#) [With the aggregation of [SwDataDefProps](#) in the role [physicalProps](#) the physical properties of the [SystemSignal](#) can be specified.]()

[TPS_SYST_05000] System Description doesn't use a complete Software Component Description [If the System Description doesn't use a complete Software Component Description (VFB View) the data mapping of [VariableDataPrototypes](#) or [ArgumentDataPrototypes](#) owned by [ClientServerOperations](#) on [SystemSignals](#) does not need to be defined. This supports the inclusion of legacy signals.] ([RS_SYST_00001](#))

[constr_3501] Role of [SystemSignal](#) in 1:n communication [In case of 1:n communication the [VariableDataPrototype](#) in the [PPortPrototype](#) of the [SwComponentPrototype](#) shall be mapped to only one [SystemSignal](#).]()

[constr_3086] Role of `SystemSignal` in n:1 sender-receiver communication [In case of n:1 communications each sender needs to be represented by the same `SystemSignal`.]()

[constr_3049] Role of `SystemSignal` in inter-ECU client server communication with clients located on different ECUs in case of networks other than Ethernet [In case of a n:1 inter-ECU client server communication with clients located on different ECUs different `SystemSignals` shall be used for each Ecu.]()

[TPS_SYST_02150] Role of `SystemSignal` in inter-ECU client server communication over Ethernet with clients located on different ECUs [In case of a n:1 inter-ECU client server communication over Ethernet with clients located on different ECUs exactly one `SystemSignal` per communication direction shall be used to define the client server interaction between the server and all clients since the relationship between the call and return is achieved by means of meta data items attached to the `Pdus` by the Socket Adapter.]()

[TPS_SYST_02151] MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs [The modeling of client server interaction over Ethernet with clients located on different ECUs requires the support of COM Stack MetaData. The relationship between the call and return is achieved by means of meta data items attached to the `Pdus` by the Socket Adapter.]()

[TPS_SYST_01087] Role of `SystemSignal` in inter-ECU client server communication with clients located on the same ECU [In case of n:1 inter-ECU client server communication it is allowed to use the same `SystemSignal` for several clients on the same Ecu, if the client identifier is used to distinguish the different clients.]()

[TPS_SYST_02011] `initValues` of receivers that are mapped to the same Ecu [All receivers of a given `SystemSignal` on the same `EcuInstance` shall have identical `initValues`.]()

[constr_3112] Invalidation support for partial mapping of a data element typed by composite data type [If a `VariableDataPrototype` with a composite data type in a `PPortPrototype` is mapped to a `SystemSignalGroup` and only a subset of elements of the composite data type that are primitives is mapped to separate `SystemSignals` of the `SystemSignalGroup` then at least one mapped primitive shall have an `invalidValue` defined.]()

[constr_3074] No `TransmissionAcknowledgementRequest` for multiple senders [If more than one `SenderComSpec` exist (in different `PortPrototypes` on atomic level) that refer to data elements effectively mapped to the same `SystemSignal` it is not allowed that any `SenderComSpec` aggregates `transmissionAcknowledge`.]()

Please note that the term “effectively mapped” refers to the fact that the `DataMapping` can refer to a `dataElement` in a “delegation” `PortPrototype` on the surface of a `rootSoftwareComposition` of an Ecu Extract OR to `PortPrototypes` inside the `rootSoftwareComposition`. Both ways shall be considered.

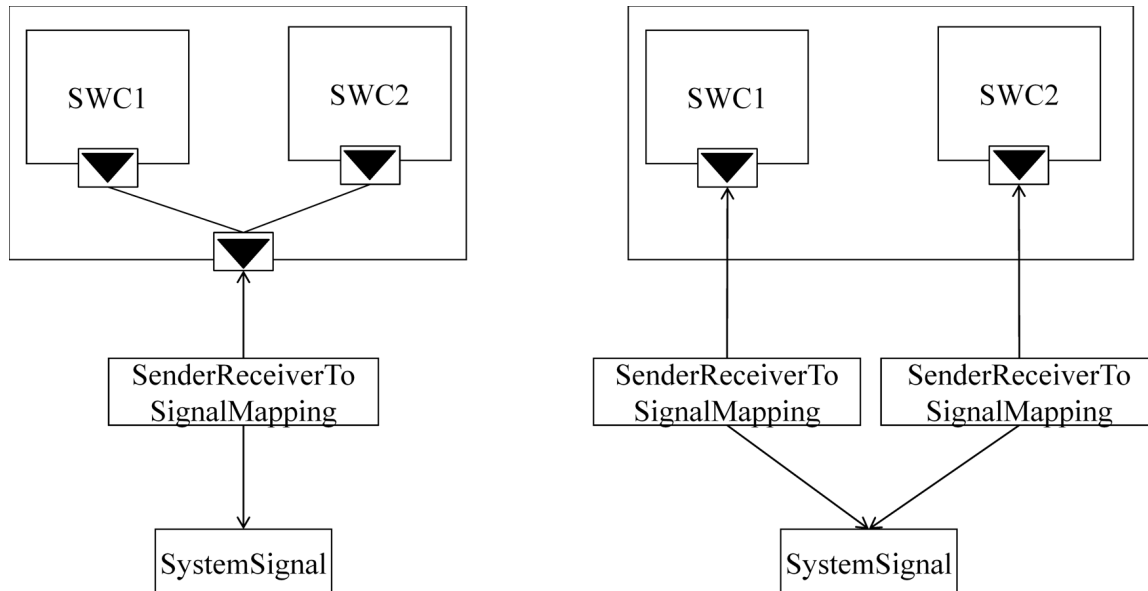


Figure 5.8: Example for data elements that are effectively mapped to the same System-Signal

The different kinds of data mapping are described in the following sections in detail.

Please note that the usage of `ImplementationDataTypes` within an `AnyInstanceRef` is described in detail in [2].

5.2.1 Mapping of Variable Data Prototypes on System Signals

This section describes how `VariableDataPrototypes` are mapped onto `System-Signals`. For a detailed description of the interconnection of software components refer to [5].

It is the task of system configuration to map `VariableDataPrototype`, `ClientServerOperation`, or `Trigger` contained in `PortPrototypes` referenced by the `SwConnector` onto a `SystemSignal`.

[TPS_SYST_01033] DataMapping and SwConnector [For the purpose of creating `DataMappings` `PortPrototypes` may or may not be connected by `SwConnectors`.]
()

The same `SystemSignal` may satisfy more than one `SwConnector` (1:n communication), and one `SwConnector` may be implemented by several `SystemSignals`

(e.g. one per `VariableDataPrototype` in the `PortInterface` being connected), so there is no 1:1 mapping between `SwConnectors` and `SystemSignals`.

In the following sections, each reference to a `VariableDataPrototype`, `ArgumentDataPrototype`, or `Trigger` is of type `AtpInstanceRef` [2]. This means it not only references the actual `VariableDataPrototype`, but additionally contains contextual references to the `PortPrototype` and the hierarchy of `SwComponentPrototypes` forming the individual instance context of the `VariableDataPrototype`.

Please note that for `DataMappings` the following use cases are supported:

- Sending: one `DataMapping` that point to the `DataPrototype` and to the `SystemSignal` with `communicationDirection` `out`.
- Receiving: one `DataMapping` that point to the `DataPrototype` and to the `SystemSignal` with `communicationDirection` `in`.
- Sending and Receiving: two `DataMappings` that point to the `DataPrototype` and to the `SystemSignal`, one with `communicationDirection` `in` and one with `communicationDirection` `out`.

In a complete `System` with `category` `SYSTEM_DESCRIPTION`, it is sufficient to refer to the `VariableDataPrototype` in the `PPortPrototype` or the `RPortPrototype` to define the mapping of the communication between a provider and its receivers.

This is possible since the connectors implicitly define which `RPortPrototype` are connected to which `PPortPrototype`. In case the `System` with `category` `SYSTEM_DESCRIPTION` does not use a complete Software Component Description (VFB View) the data mapping needs not to be defined. This supports the inclusion of legacy signals.

[TPS_SYST_01137] Several `DataMappings` may be defined for the same `SystemSignal` [For a `SystemSignal` which is

- part in several `SystemSignalGroups`
- part in at least one `SystemSignalGroup` and at the same time is transmitted additionally as standalone `SystemSignal`

several `DataMappings` may be defined.]()

[TPS_SYST_01050] `SystemSignal` in the System Extract and ECU Extract [In the `System` with `category` `SYSTEM_EXTRACT` or `ECU_EXTRACT` the missing `DataMappings` on the complementary Sender/Receiver side needs to be supplemented.]()

In the `System` with `category` `SYSTEM_EXTRACT` or `ECU_EXTRACT`, where only the relevant parts of the `rootSoftwareComposition` are defined, it is necessary to utilize the information from the complementary `PortPrototype` if the corresponding `PortPrototype` is located on another ECU and thus is not part of the extract. This is described in more detail in chapter 11.2 and chapter 12.2.3.

Therefore in a `System` with `category ECU_EXTRACT` the `DataMappings` are provided on both, `PPortPrototypes` and `RPortPrototypes`.

[TPS_SYST_01034] Data Mappings can be applied to compositions and atomic software components [`DataMappings` can be applied to `CompositionSwComponentTypes` and on `AtomicSwComponentTypes`.]()

[TPS_SYST_01035] Transformation of Data Mappings during flattening [During the creation of the `System` with `category ECU_EXTRACT` (flattening) the existing `DataMappings` that refer to `CompositionSwComponentTypes` shall be transformed to refer to `AtomicSwComponentTypes` instead.]()

[TPS_SYST_01036] No additional Data Mappings in composition substructure [When a `CompositionSwComponentType` is refined by a supplier the already existing `DataMappings` that refer to the `CompositionSwComponentType` shall not be copied to the internal substructure.]()

Suppliers who add substructure to a `CompositionSwComponentType` by adding `SwComponentPrototypes` and `SwConnectors` shall respect the predefined `DataMappings` on the `CompositionSwComponentType`.

The OEM/Supplier Collaboration Scenario is described in chapter 11.1.

Please note that for `DataMappings` the following use cases are supported:

- Sending: one `DataMapping` that point to the `DataPrototype` and to the `SystemSignal` with `communicationDirection out`.
- Receiving: one `DataMapping` that point to the `DataPrototype` and to the `SystemSignal` with `communicationDirection in`.
- Sending and Receiving: two `DataMappings` that point to the `DataPrototype` and to the `SystemSignal`, one with `communicationDirection in` and one with `communicationDirection out`.

[constr_1207] Existence of the attribute `DataMapping.communicationDirection` in the context of a `SenderReceiverInterface` or `TriggerInterface`

[The following condition shall be fulfilled regarding the existence and values of the attribute `DataMapping.communicationDirection` that refers to a `PortPrototype` typed by a `SenderReceiverInterface` or `TriggerInterface` as the context `PortPrototype`:

- If the `DataMapping` refers to a `PPortPrototype` as the context `PortPrototype` the attribute `DataMapping.communicationDirection` may exist. If the attribute exists its value shall be set to `out`.
- If the `DataMapping` refers to an `RPortPrototype` as the context `PortPrototype` the attribute `DataMapping.communicationDirection` may exist. If the attribute exists its value shall be set to `in`.

]()

[TPS_SYST_05034] **DataMapping** of **ImplementationDataType** of category **UNION, DATA_REFERENCE, or FUNCTION_REFERENCE** [`SenderReceiverInterface.dataElement` that is typed by an `ImplementationDataType` of category **UNION, DATA_REFERENCE, or FUNCTION_REFERENCE** is not allowed to be mapped by a `DataMapping`.]()

Please note that for unions there is [constr_1441] and [constr_1607] that are restricting such a modeling.

5.2.1.1 Mapping of Variable Data Prototypes with primitive datatypes on System Signals (Sender-Receiver Communication)

This section describes the relation between the `VariableDataPrototype` with primitive datatypes and the `SystemSignal` (see Figure 5.9).

[TPS_SYST_02082] **SenderReceiverInterface.dataElement** is typed by an **ApplicationPrimitiveDataType** of category **VALUE** or **BOOLEAN** and a **DataTypeMap** exists [If a `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category **VALUE** or **BOOLEAN** and a `DataTypeMap` exists that points to the `ApplicationPrimitiveDataType` and an `ImplementationDataType` of category **VALUE** or **TYPE_REFERENCE** that eventually references (via the `SwDataDefProps.implementationDataType`) an `ImplementationDataType` of category **VALUE**, then this `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02083] **SenderReceiverInterface.dataElement** is typed by an **ApplicationPrimitiveDataType** of category **STRING** and a **DataTypeMap** exists [If a `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category **STRING** and a `DataTypeMap` exists that points to the `ApplicationPrimitiveDataType` and an `ImplementationDataType` of category **ARRAY** or **TYPE_REFERENCE** that eventually references (via the `SwDataDefProps.implementationDataType`) an `ImplementationDataType` of category **ARRAY** with a `subElement` that either

- represents the platform type `uint8` or
- references a `SwBaseType` with a `SwBaseType.baseTypeDefinition.baseTypeSize` set to the value 8 and the `SwBaseType.baseTypeDefinition.baseTypeEncoding` set to `NONE`,

then this `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02084] **SenderReceiverInterface.dataElement** is typed by an **ApplicationArrayDataType** and a **DataTypeMap** exists [If a `SenderReceiverInterface.dataElement` is typed by an `ApplicationArrayDataType` and a `DataTypeMap` exists that points to the `ApplicationArrayDataType` and an

`ImplementationDataType` of category `ARRAY` or `TYPE_REFERENCE` that eventually references (via the `SwDataDefProps.implementationDataType`) an `ImplementationDataType` of category `ARRAY` with a `subElement` that either

- represents the platform type `uint8` or
- references a `SwBaseType` with a `SwBaseType.baseTypeDefinition.baseTypeSize` set to the value `8` and the `SwBaseType.baseTypeDefinition.baseTypeEncoding` set to `NONE`,

then this `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02085] `SenderReceiverInterface.dataElement` is typed by an `ImplementationDataType` of category `ARRAY` [If a `SenderReceiverInterface.dataElement` is typed by an `ImplementationDataType` of category `ARRAY` with a `subElement` that either

- represents the platform type `uint8` or
- references a `SwBaseType` with a `SwBaseType.baseTypeDefinition.baseTypeSize` set to the value `8` and the `SwBaseType.baseTypeDefinition.baseTypeEncoding` set to `NONE`,

then this `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02086] `SenderReceiverInterface.dataElement` is typed by an `ImplementationDataType` of category `VALUE` or `TYPE_REFERENCE` [If a `SenderReceiverInterface.dataElement` is typed by an `ImplementationDataType` of category `VALUE` or `TYPE_REFERENCE` that eventually references (via the `SwDataDefProps.implementationDataType`) an `ImplementationDataType` of category `VALUE` then this `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02087] `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category `BOOLEAN` and no `DataTypeMap` exists [The `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02088] `SenderReceiverInterface.dataElement` is typed by an `ApplicationArrayDataType` and no `DataTypeMap` exists [If a `SenderReceiverInterface.dataElement` is typed by an `ApplicationArrayDataType` and no `DataTypeMap` exists and the `ApplicationArrayDataType` fulfills the following conditions:

- `ApplicationPrimitiveDataType.swDataDefProps.dataConstr` exists and refers to a `PhysConstrs`.
- `ApplicationPrimitiveDataType.swDataDefProps.compuMethod` exists and refers to a `CompuMethod` of category `TEXTTABLE` and `CompuMethod.compuPhysToInternal` exists.

- Application of `ApplicationPrimitiveDataType.swDataDefProps.compuMethod` to `ApplicationPrimitiveDataType.swDataDefProps.dataConstr` yields a numerical range in [0 .. 255]

then the `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02089] `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category `STRING` and no `DataTypeMap` exists [If a `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category `STRING` and no `DataTypeMap` exists and the `ApplicationPrimitiveDataType` fulfills the following conditions:

- `ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout` exists and values of `SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupFrom` and `SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupTo` are both set to 1.
- `ApplicationPrimitiveDataType.swDataDefProps.swTextProps` exists and refers to an `SwBaseType` where the `SwBaseType.baseTypeDefinition.baseTypeEncoding` is set to `NONE` and the value of `SwBaseType.baseTypeDefinition.baseTypeSize` is set to 8.

then the `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_02090] `SenderReceiverInterface.dataElement` is typed by an `ApplicationPrimitiveDataType` of category `VALUE` and no `DataTypeMap` exists [There is no clear indication that the `SenderReceiverInterface.dataElement` is a candidate for a primitive Data Mapping.]()

[TPS_SYST_01037] primitive Data Mapping of `UINT8-Arrays` [The primitive Data Mapping may also be used for the Data Mapping of `UINT8-Arrays`. This supports an optimized definition of the Data Mapping.]()

In other words it is allowed to map an array `VariableDataPrototype` consisting of `UINT8` elements to exactly one `SystemSignal` in the context of one `SenderReceiverToSignalMapping`. A `UINT8` element may be a String or an array that contains array elements of Integer type with range 0..255.

Background: In the ECU Configuration of the AUTOSAR COM module such a `SystemSignal` will be mapped to a COM Signal with the `ComSignalType` `UINT8_N`.

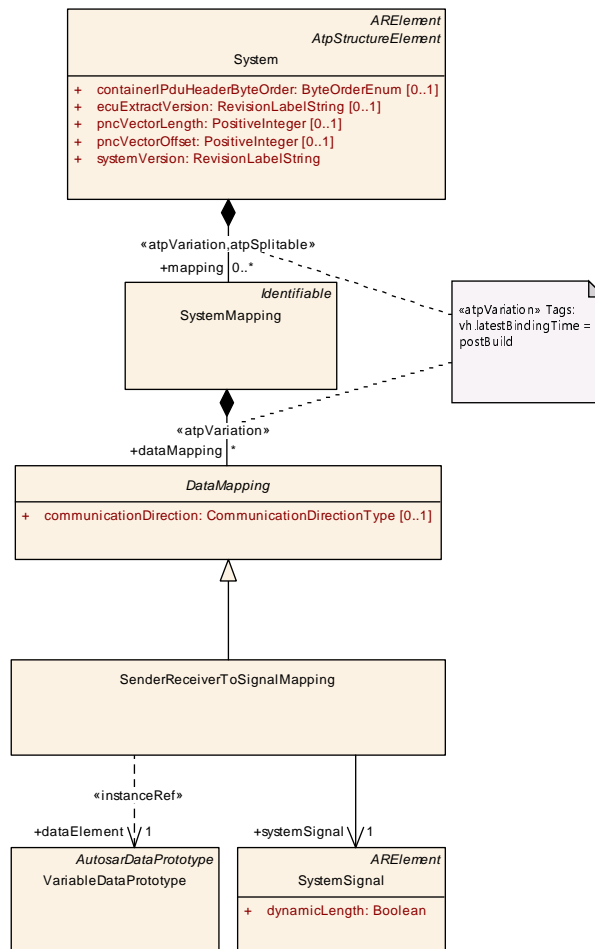


Figure 5.9: Mapping of data elements with primitive datatypes (SenderRecPrimitiveTypeMapping)

Class	SenderReceiverToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element to a signal.			
Base	<i>ARObject</i> , <i>DataMapping</i>			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	1	iref	Reference to the data element.
systemSignal	SystemSignal	1	ref	Reference to the system signal used to carry the data element.

Table 5.16: SenderReceiverToSignalMapping

5.2.1.2 Mapping of Variable Data Prototypes with composite datatypes (Sender-Receiver Communication)

This section describes the mapping of [VariableDataPrototypes](#) typed by composite data types to [SystemSignals](#).

It is not possible to map a [VariableDataPrototype](#) typed by composite data type directly (without any additional mechanisms) to one [SystemSignal](#) because The RTE is required to treat AUTOSAR signals transmitted using sender-receiver communication consistently. For this purpose, data transformation or [SystemSignalGroups](#) is used.

There are two ways to map a [VariableDataPrototype](#) typed by composite data type to [SystemSignals/SystemSignalGroups](#):

1. Use data transformation and map it directly to a [SystemSignal](#).
2. Map it to a [SystemSignalGroup](#) with [SenderReceiverToSignalGroupMapping](#)

[constr_3506] Mapping of composite data type to [SystemSignals](#) in [SystemSignalGroup](#) [Either all or a subset of elements of a composite data type shall be mapped to [SystemSignals](#) which shall be members of one [SystemSignalGroup](#) if no data transformation (except COM Based Transformer) is used.

There are two exceptions to this rule:

- it is allowed to map an array [VariableDataPrototype](#) consisting of `UINT8` elements to exactly one [SystemSignal](#) in the context of one [SenderReceiverToSignalMapping](#) (see [TPS_SYST_01037]).
- in case the COM Based Transformer [17] is used it is the integral part of the approach to have a fixed mapping of the individual elements of composite data types to [SystemSignals](#) in a [SystemSignalGroup](#) ([TPS_SYST_02058]).

]() 1

5.2.1.2.1 Data Transformation

If data transformation is used, the consistency of the composite data is assured by the transformation.

A `VariableDataPrototype` typed by composite data type can be mapped to one `SystemSignal` without any `SystemSignalGroup` if data transformation is used.

In that case any required mapping between the `ApplicationCompositeElementDataPrototypes` of the `VariableDataPrototype` of the connected `PortPrototypes` needs to be expressed by means of a `PortInterfaceMapping` attached to the `SwConnector` connecting the two `PortPrototypes` and not by means of two separated `DataMappings` (one referencing the `VariableDataPrototype` at the `PPortPrototype` and the other one referencing the `VariableDataPrototype` at the `RPortPrototype`).

During creation of a System Extract of the System Configuration Description or the creation of an ECU Extract of the System Configuration Description, this `PortInterfaceMapping` needs to be preserved in order to support proper deserializing transformation at the receiver side (see chapter 11.4 and 12.2).

See chapter 7 for details how to enable data transformation.

In case the COM Based Transformer [17] is used the mapping from section 7.3.3 is required.

5.2.1.2.2 Mapping via SystemSignalGroups

The `VariableDataPrototype` that is referenced by `dataElement` can be typed by an `ApplicationDataType` or by an `ImplementationDataType`. This type decides which reference is used within the `SenderRecRecordElementMapping` and `SenderRecArrayElementMapping`.

Composite `DataPrototypes` may nest within composite `VariableDataPrototypes`. Each element typed by a primitive data type of such nested composite `VariableDataPrototype` can be mapped to one `SystemSignal` of a `SystemSignalGroup`.

Please note that not every single element typed by a primitive data type needs to be mapped to a `SystemSignal` since a partial mapping is also supported that maps a subset of elements of the composite data type to separate `SystemSignals` of a `SystemSignalGroup`.

The mapping between the `SystemSignal` and the `VariableDataPrototype` is provided in the `SenderReceiverToSignalGroupMapping` (see Figure 5.10).

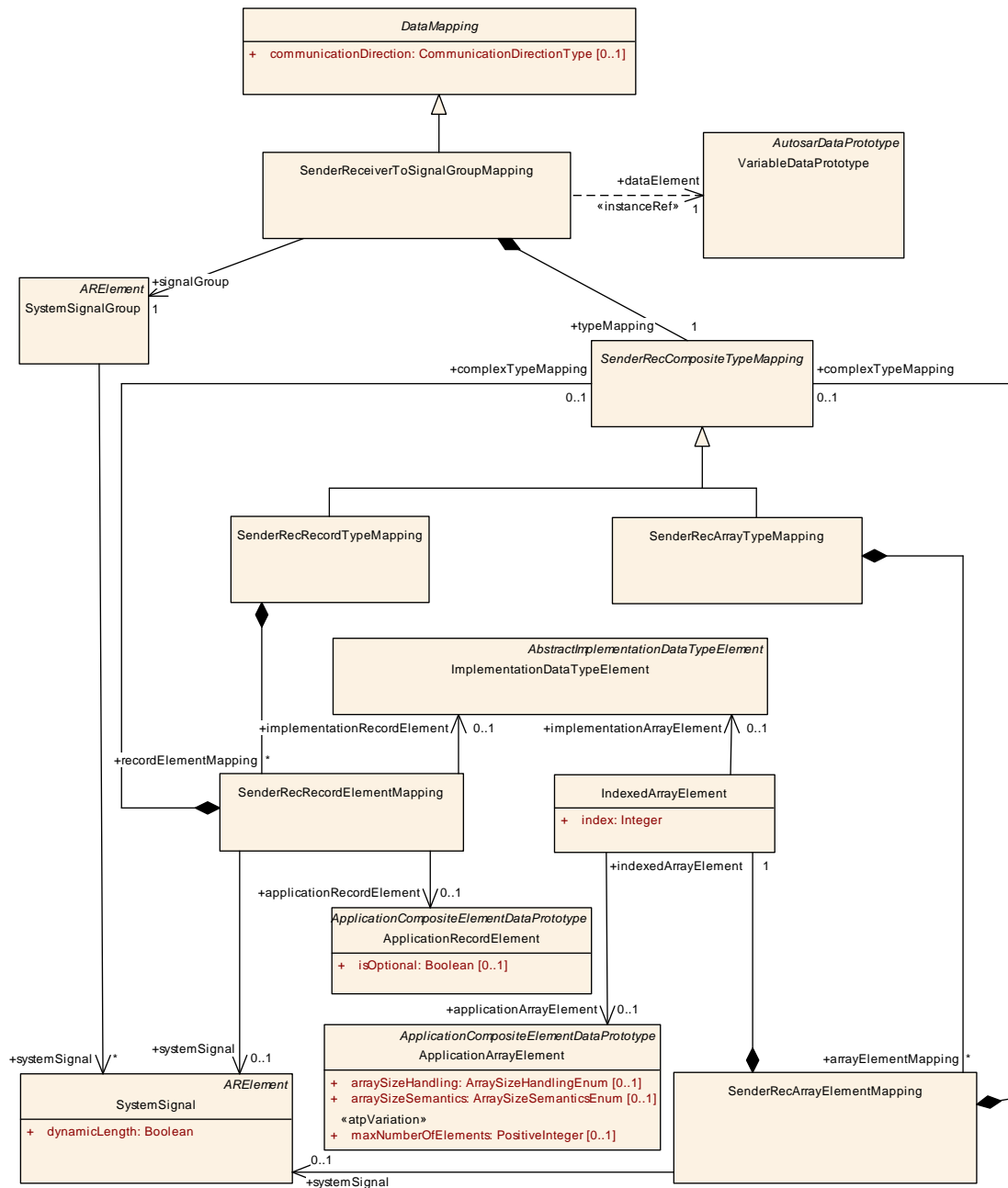


Figure 5.10: Mapping of data elements with composite data types (SenderRecCompositeTypeMapping)

[constr_3000] valid SenderRecCompositeTypeMappings [SenderReceiverToSignalGroupMapping.signalGroup.systemSignal shall point to each SystemSignal being mapped within the context of SenderReceiverToSignalGroupMapping.

In other words: For each SystemSignal referenced in the role SenderReceiverToSignalGroupMapping.signalGroup.systemSignal there shall be either a reference in the role SenderRecRecordElementMapping.systemSignal or a reference in the role SenderRecArrayElementMapping.systemSignal aggregated by

the same [SenderReceiverToSignalGroupMapping](#) that refers to this [System-Signal.>\(\)](#)

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			
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.
signalGroup	SystemSignalGroup	1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type
typeMapping	SenderRecCompositeTypeMapping	1	aggr	The CompositeTypeMapping maps the ApplicationArray Elements and ApplicationRecordElements to Signals of the SignalGroup.

Table 5.17: SenderReceiverToSignalGroupMapping

Class	SenderRecCompositeTypeMapping (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	<p>Two mappings exist for the composite data types: "ArrayTypeMapping" and "RecordTypeMapping". In both, a primitive datatype will be mapped to a system signal.</p> <p>But it is also possible to combine the arrays and the records, so that an "array" could be an element of a "record" and in the same manner a "record" could be an element of an "array". Nesting these data types is also possible.</p> <p>If an element of a composite data type is again a composite one, the "CompositeTypeMapping" element will be used one more time (aggregation between the ArrayElementMapping and CompositeTypeMapping or aggregation between the RecordElementMapping and CompositeTypeMapping).</p>			
Base	ARObject			
Subclasses	SenderRecArrayTypeMapping , SenderRecRecordTypeMapping			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 5.18: SenderRecCompositeTypeMapping

Class	SenderRecArrayTypeMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is an Array, the "ArrayTypeMapping" will be used.			
Base	ARObject, SenderRecCompositeTypeMapping			
Attribute	Type	Mult.	Kind	Note
arrayElement Mapping	SenderRecArrayElementMapping	*	aggr	Each ApplicationArrayElement shall be mapped on a SystemSignal.

Table 5.19: SenderRecArrayTypeMapping

Class	SenderRecRecordTypeMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is a Record, the "RecordTypeMapping" will be used.			
Base	ARObject, SenderRecCompositeTypeMapping			
Attribute	Type	Mult.	Kind	Note
recordElement Mapping	SenderRecRecordElementMapping	*	aggr	Each ApplicationRecordElement shall be mapped on a SystemSignal.

Table 5.20: SenderRecRecordTypeMapping

Class	SenderRecRecordElementMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	<p>Mapping of a primitive record element to a SystemSignal. If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference application RecordElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference implementationRecordElement shall be used. Either the implementationRecordElement or applicationRecordElement reference shall be used.</p> <p>If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the RecordElementMapping element will aggregate the complexTypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
application RecordElement	ApplicationRecordElement	0..1	ref	Reference to an ApplicationRecordElement in the context of the dataElement or in the context of a composite element.
complexType Mapping	SenderRecCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
implementation RecordElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationRecordElement in the context of the dataElement or in the context of a composite element.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement.

Table 5.21: SenderRecRecordElementMapping

[constr_3230] Usage of [SenderRecRecordElementMapping.applicationRecordElement](#) [[SenderRecRecordElementMapping.applicationRecordElement](#) shall only be used if the referenced context element ([VariableDataPrototype](#) that is referenced by the [SenderReceiverToSignalGroupMapping.dataElement](#)) is typed by an [ApplicationDataType](#).]()

[constr_3244] Usage of [SenderRecRecordElementMapping.implementationRecordElement](#) [[SenderRecRecordElementMapping.implementationRecordElement](#) shall only be used if the referenced context element ([VariableDataPrototype](#) that is referenced by the [SenderReceiverToSignalGroupMapping.dataElement](#)) is typed by an [ImplementationDataType](#).]()

Class	SenderRecArrayElementMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	<p>The SenderRecArrayElement may be a primitive one or a composite one. If the element is primitive, it will be mapped to the SystemSignal (multiplicity 1). If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference to the ApplicationArrayElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference to the ImplementationArrayElement shall be used.</p> <p>If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the ArrayElementMapping element will aggregate the TypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals.</p> <p>Regardless whether composite or primitive array element is mapped the indexed element always needs to be specified.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
complexTypeMapping	SenderRecCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
indexedArrayElement	IndexedArrayElement	1	aggr	Reference to an indexed array element in the context of the dataElement or in the context of a composite element.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationArrayElement.

Table 5.22: SenderRecArrayElementMapping

Class	IndexedArrayElement			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element represents exactly one indexed element in the array. Either the applicationArrayElement or implementationArrayElement reference shall be used.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
applicationArrayElement	ApplicationArrayElement	0..1	ref	Reference to an ApplicationArrayElement in an array.
implementationArrayElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array.
index	Integer	1	attr	Position of an element in an array. Starting position is 0.

Table 5.23: IndexedArrayElement

[constr_3231] Usage of [IndexedArrayElement.applicationArrayElement](#) [[IndexedArrayElement.applicationArrayElement](#) shall only be used if the referenced context element ([VariableDataPrototype](#) that is referenced by the [SenderReceiverToSignalGroupMapping.dataElement](#)) is typed by an [ApplicationDataType](#).]()

[constr_3245] Usage of `IndexedArrayElement.implementationArrayElement` [`IndexedArrayElement.implementationArrayElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping.dataElement`) is typed by an `ImplementationDataType`.]()

Figure 5.11 shows a mapping example for nested composite data types.

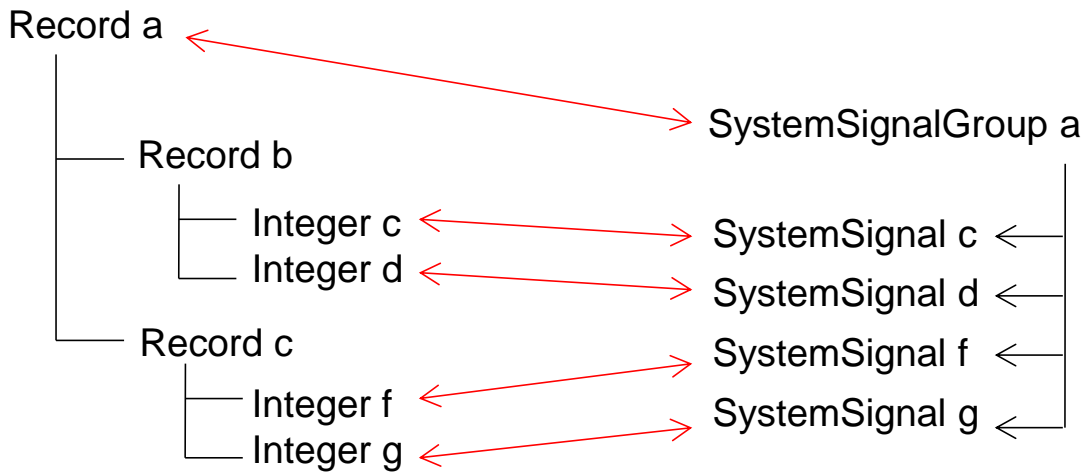


Figure 5.11: Mapping example for nested composite data types

Record a is mapped with `SenderReceiverToSignalGroupMapping` to a `SystemSignalGroup`. The content of *Record a* is mapped with the `SenderRecRecordTypeMapping`. Since the first element of *Record a* is *Record b* the `SenderRecRecordElementMapping` does not contain a reference to a `SystemSignal` because signals apply only to atomic data items. Instead it contains a `complexTypeMapping` with two `SenderRecRecordElementMappings` for *Integer c* and *Integer d*. These two elements are mapped to `SystemSignals`.

Please note that a partial mapping of a data element typed by composite data type in a `PPortPrototype` is also supported. If a `VariableDataPrototype` with a composite data type in a `PPortPrototype` is mapped to a `SystemSignalGroup` then it is allowed to map only a subset of elements of the composite data type that are primitives to separate `SystemSignals` of the `SystemSignalGroup`. This means that it is possible to transmit a subset of a composite data element in a `ISignalGroup` over the network. Figure 5.12 shows a partial mapping example for nested composite data types.

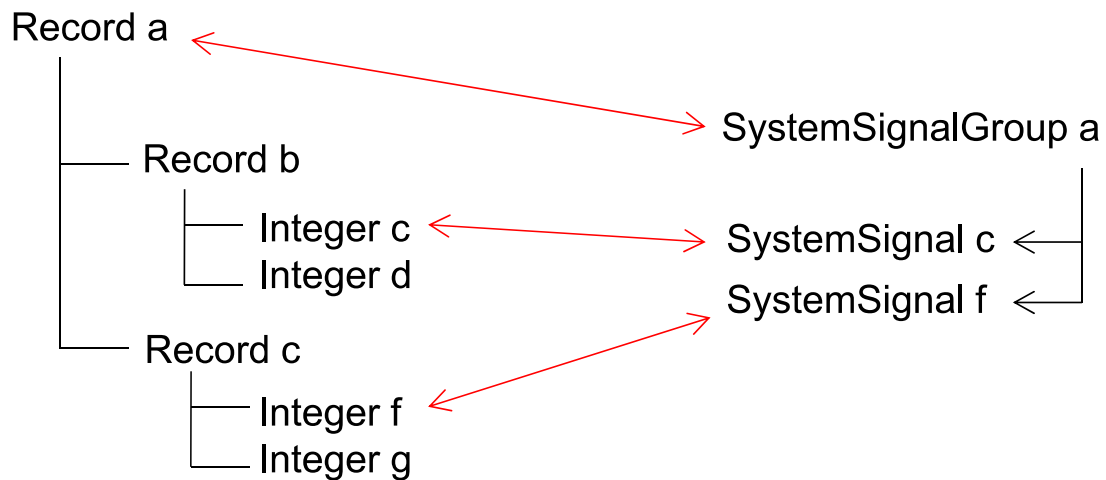


Figure 5.12: Partial mapping example for nested composite data types

5.2.1.3 Mapping of Client Server Operations to System Signals

This section describes the mapping of `ClientServerOperations` to `SystemSignals` (see Figure 5.13).

[TPS_SYST_01148] Mapping of IN and INOUT `ArgumentDataPrototypes` to `callSignals` [The `ArgumentDataPrototypes` that are passed to the operation (i.e. the `direction` is “in”) and the `ArgumentDataPrototypes` that are passed to and returned from the operation (i.e. the `direction` is “inout”) are expected to be mapped to the `callSignal` by the serializer.]()

[TPS_SYST_01149] Mapping of OUT and INOUT `ArgumentDataPrototypes` to `returnSignals` [The `ArgumentDataPrototypes` that are returned from the operation (i.e. the `direction` is “out”) and the `ArgumentDataPrototypes` that are passed to and returned from the operation (i.e. the `direction` is “inout”) are expected to be mapped to the `returnSignal` by the serializer.]()

[TPS_SYST_01150] Mapping of `returnSignal` and `callSignal` to COM Signal [In the ECU Configuration of the AUTOSAR COM module the `returnSignal` and the `callSignal` are expected to be mapped to COM Signals with the `ComSignalType` `UINT8_N` or `UINT8_DYN`.]()

The `ClientServerToSignalMapping` can only map transformed data to `SystemSignals` because it contains no information how data shall be serialized, it only references the primitive `SystemSignal` which shall contain the serialized data. How to define the necessary information which serialization algorithm shall be applied can be found in chapter 7. The implementation of this algorithm is provided via a BSW module.

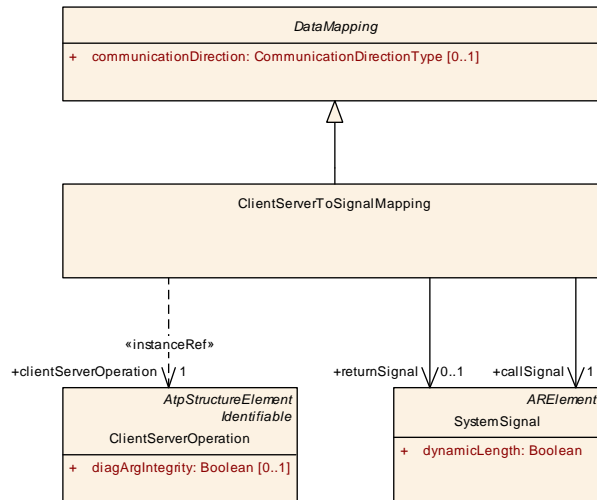


Figure 5.13: Mapping of a ClientServerOperation to a callSignal and a returnSignal

Class	ClientServerToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals.			
Base	ARObject, DataMapping			
Attribute	Type	Mult.	Kind	Note
callSignal	SystemSignal	1	ref	Reference to the callSignal to which the IN and INOUT ArgumentDataPrototypes are mapped.
clientServer Operation	ClientServerOperation	1	iref	Reference to a ClientServerOperation, which is mapped to a call SystemSignal and a return SystemSignal.
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped. Tags: atp.Status=shallBecomeMandatory

Table 5.24: ClientServerToSignalMapping

[constr_3111] returnSignal in ClientServerToSignalMapping is mandatory
 [A ClientServerToSignalMapping shall always have a returnSignal defined.]
 ()

[constr_3215] TransformationTechnology.version and TransformationTechnology.protocol settings for request and response of a client/server communication
 [TransformationTechnology.version and TransformationTechnology.protocol shall be identical for ISignals that are derived from the same ClientServerOperation. This means that all ISignals that refer to ClientServerToSignalMapping.callSignal or to ClientServerToSignalMapping.returnSignal of the same ClientServerToSignalMapping shall have the same TransformationTechnology.protocol and TransformationTechnology.version defined.]()

5.2.1.4 Mapping of a ApplicationCompositeElementDataPrototype within a composite application data type on a System Signal (Sender-Receiver Communication)

SenderReceiverCompositeElementToSignalMapping is used to map a ApplicationCompositeElementDataPrototype that is aggregated within a composite data type (record element or an array element) to a SystemSignal.

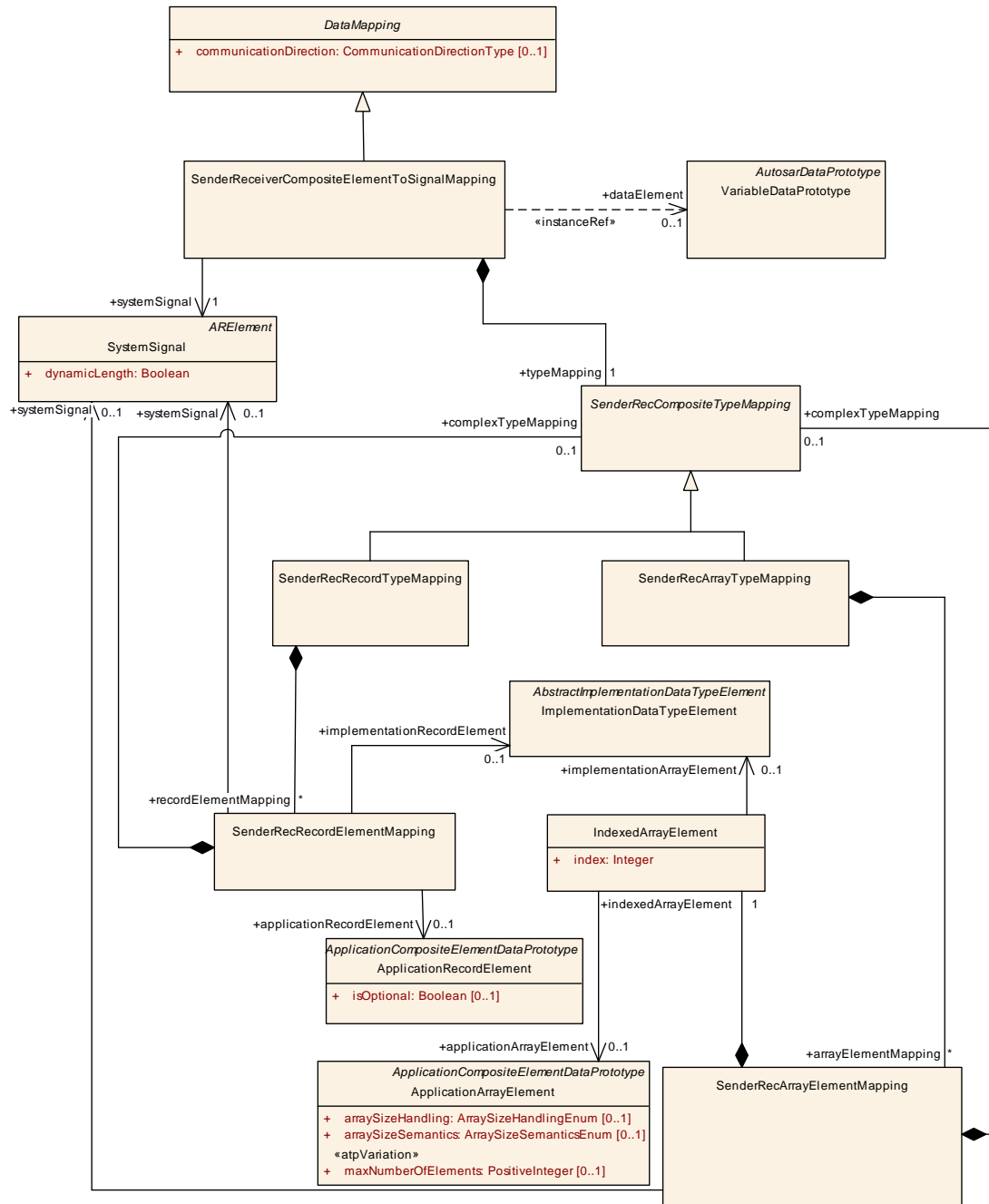


Figure 5.14: Mapping of a Variable Data Prototype which is aggregated within a composite data type on a System Signal

[constr_3058] References from [SenderRecArrayElementMapping](#) and from [SenderRecRecordElementMapping](#) to [SystemSignals](#) are not allowed within a [SenderReceiverCompositeElementToSignalMapping](#) [The reference from [SenderRecArrayElementMapping](#) to [SystemSignal](#) and from [SenderRecRecordElementMapping](#) to [SystemSignal](#) shall not exist if the enclosing [SenderRecCompositeTypeMapping](#) is owned by a [SenderReceiverCompositeElementToSignalMapping](#).]()

[TPS_SYST_01143] [DataMapping](#) on the sender side for elements of a composite data type [On the sender side, it is possible that only a subset of elements of an [ApplicationCompositeElementDataPrototype](#) of a [dataElement](#) in a [PPortPrototype](#) in its sender role is referenced by a [DataMapping](#).]()

Class	SenderReceiverCompositeElementToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of an Variable Data Prototype which is aggregated within a composite datatype to a System Signal (only one element of the composite data type is mapped).			
Base	ARObject , DataMapping			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	0..1	iref	Reference to a data element with a composite datatype from which one element is mapped to a SystemSignal.
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal to which one primitive of the composite type is mapped.
typeMapping	SenderRecCompositeTypeMapping	1	aggr	The CompositeTypeMapping maps one VariableData Prototype of the composite data type to a SystemSignal.

Table 5.25: SenderReceiverCompositeElementToSignalMapping

[SenderRecCompositeTypeMapping](#) and all subclasses are described in section [5.2.1.2](#)

5.2.1.5 Mapping of Trigger to SystemSignal

[TPS_SYST_05001] Send a [Trigger](#) across a network [In order to be able to send a [Trigger](#) across a network to trigger a [RunnableEntity](#) deployed to a different [EcuInstance](#) it is possible to define a [TriggerToSignalMapping](#) that maps a [Trigger](#) to a [SystemSignal](#) in the role [systemSignal](#).]()

[constr_1198] [TriggerToSignalMapping.systemSignals](#) eligible for a [TriggerToSignalMapping](#) [In the context of a [TriggerToSignalMapping](#), it is only possible to refer to a [TriggerToSignalMapping.systemSignal](#) that in turn is referenced by an [ISignal](#) with attribute [length](#) set to 0.]()

[constr_1199] [ISignals](#) relating to [systemSignals](#) eligible for a [TriggerToSignalMapping](#) [An [ISignal](#) used to reference a [systemSignal](#) that in turn is referenced by a [TriggerToSignalMapping](#) shall also be referenced by an [ISignalToIPduMapping](#) where the attribute [updateIndicationBitPosition](#) is defined.]()

[TPS_SYST_05002] The value of **startPosition** is irrelevant [The value of **startPosition** shall not be considered inside an **ISignalToIPduMapping** that references an **ISignal** used to reference a **TriggerToSignalMapping.systemSignal** that in turn is referenced by a **TriggerToSignalMapping**.]()

[constr_3065] Mapping of queued **Triggers** to **SystemSignals** is prohibited [A **TriggerToSignalMapping** of a **Trigger** with **swImplPolicy** set to **queued** is prohibited.]()

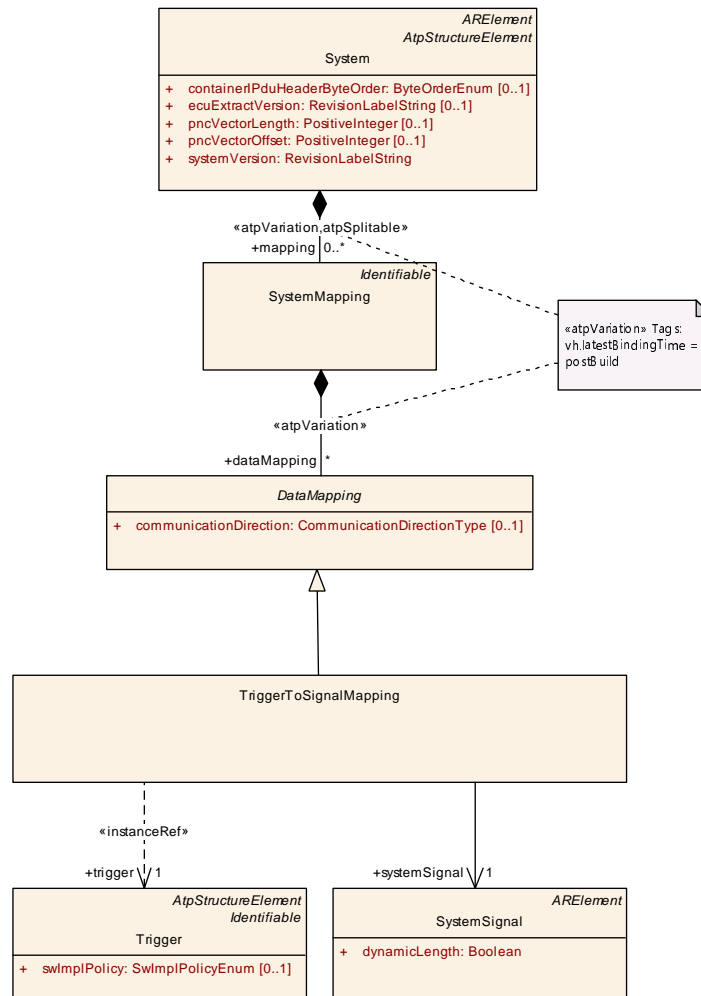


Figure 5.15: Structure of a **TriggerToSignalMapping**

Class	TriggerToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This meta-class represents the ability to map a trigger to a SystemSignal of size 0. The Trigger does not transport any other information than its existence, therefore the limitation in terms of signal length.			
Base	ARObject, DataMapping			
Attribute	Type	Mult.	Kind	Note





Class	TriggerToSignalMapping			
systemSignal	SystemSignal	1	ref	This is the SystemSignal taken to transport the Trigger over the network. Tags: xml.sequenceOffset=20
trigger	Trigger	1	iref	This represents the Trigger that shall be used to trigger RunnableEntities deployed to a remote ECU. Tags: xml.sequenceOffset=10

Table 5.26: TriggerToSignalMapping

5.2.2 Signal Path Constraint

One task of the System Generator is to define the needed communication infrastructure (e.g. [ISignals](#), [Pdus](#), [Frames](#)) between ECUs. The System Generator often has the choice between alternative paths through the topology. In the example shown in Figure 5.16 the System Generator would have the choice between two paths (Path1: CAN3 or Path2: CAN1-GW-CAN2) for a signal that is send by ECU2 and is received by ECU4. If no further information is given the decision will be made e.g. by means of boundary conditions like busload, transmissions speed, etc.

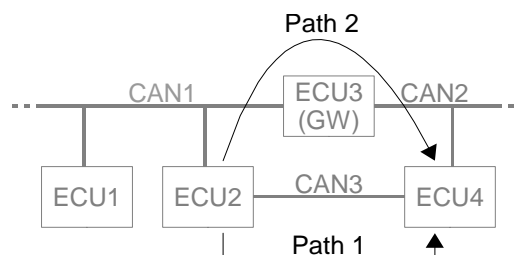


Figure 5.16: Example for a Communication Path

Signal Mapping Constraints allow to further restrict or specify the path(s) a signal is allowed to be transmitted over. A path is specified by an list of [PhysicalChannels](#).

There exist four different constraints for signals regarding the signal path (see Figure 5.17):

[TPS_SYST_01041] CommonSignalPath definition [The [CommonSignalPath](#) describes that two or more signals shall take the same path in the topology.] ([RS_SYST_00017](#))

[TPS_SYST_01042] ForbiddenSignalPath definition [The [ForbiddenSignalPath](#) describes the path that one or more signals shall not take in the topology, e.g. in case of safety critical transmission.] ([RS_SYST_00020](#))

[TPS_SYST_01043] PermissibleSignalPath definition [The [PermissibleSignalPath](#) describes the path one or more signals may take in the topology. If more than one [PermissibleSignalPath](#) is defined for the same signal/operation attributes, any of them may be chosen.] ([RS_SYST_00019](#), [RS_SYST_00016](#))

[TPS_SYST_01044] SeparateSignalPath definition [The [SeparateSignalPath](#) describes that two or more signals shall take separate paths in the topology e.g. in case of redundant transmission.] ([RS_SYST_00018](#))

It is also possible that the same signal is aggregated two times by the [SeparateSignalPath](#) element to indicate that this signal should be transmitted redundantly over two different paths.

The meta-model part, which describes the Communication Path constraints, will be explained in the following sections.

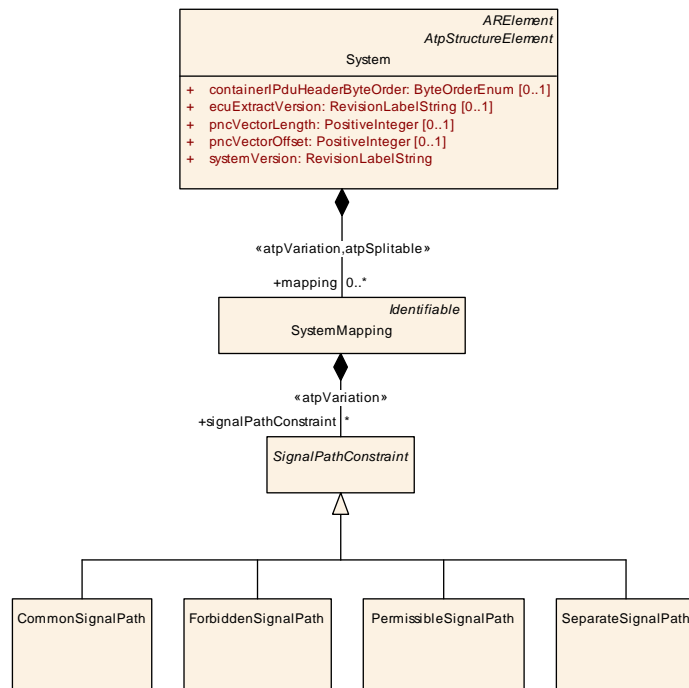


Figure 5.17: Communication Path Description (SignalPathConstraints)

5.2.2.1 CommonSignalPath

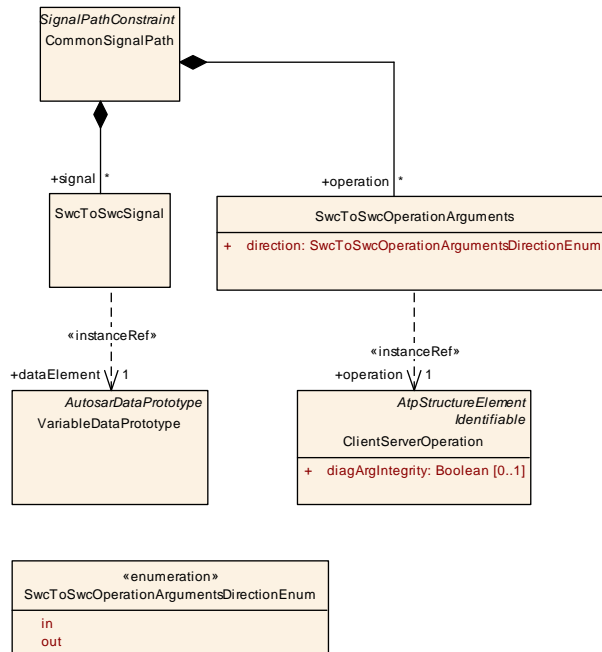


Figure 5.18: Description of signals that shall take the same way in the topology (CommonSignalPath)

Class	CommonSignalPath			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The CommonSignalPath describes that two or more SwcToSwcSignals and/or SwcToSwcOperation Arguments shall take the same way (Signal Path) in the topology.			
Base	<i>ARObject</i> , <i>SignalPathConstraint</i>			
Attribute	Type	Mult.	Kind	Note
operation	SwcToSwcOperation Arguments	*	aggr	The arguments sent in one direction (either from client to server or server to client) of the operations that shall take the same signal path.
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that shall take the same way (Signal Path) in the topology.

Table 5.27: CommonSignalPath

Class	SwcToSwcSignal			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SwcToSwcSignal describes the information (data element) that is exchanged between two SW Components. On the SWC Level it is possible that a SW Component sends one data element from one P-Port to two different SW Components (1:n Communication). The SwcToSwcSignal describes exactly the information which is exchanged between one P-Port of a SW Component and one R-Port of another SW Component.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	2	iref	Reference to a data element on the PPortPrototype and to the same data element on the RPortPrototype.

Table 5.28: SwcToSwcSignal

Class	SwcToSwcOperationArguments			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SwcToSwcOperationArguments describes the information (client server operation arguments, plus the operation identification, if required) that are exchanged between two SW Components from exactly one client to one server, or from one server back to one client. The direction attribute defines which direction is described. If direction == IN, all arguments sent from the client to the server are described by the SwcToSwcOperationArguments, in direction == OUT, it's the arguments sent back from server to client.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
direction	SwcToSwcOperationArgumentsDirectionEnum	1	attr	Direction addressed by this SwcToSwcClientServer Operation element.
operation	ClientServerOperation	2	iref	Reference to the operation at the client and at the server side whose arguments are described by SwcToSwcOperationArguments. The two ports referenced shall be connected by a connector in the software component description.

Table 5.29: SwcToSwcOperationArguments

Enumeration	SwcToSwcOperationArgumentsDirectionEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths
Note	Direction addressed by this element.
Literal	Description
in	IN (all IN and INOUT arguments) Tags: atp.EnumerationLiteralIndex=0
out	OUT (all OUT and INOUT arguments) . Tags: atp.EnumerationLiteralIndex=1

Table 5.30: SwcToSwcOperationArgumentsDirectionEnum

5.2.2.2 ForbiddenSignalPath

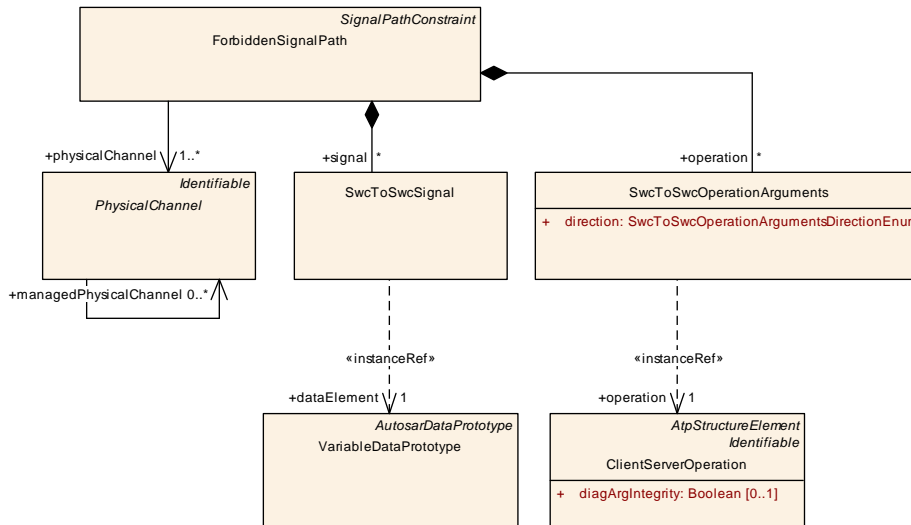


Figure 5.19: Description of the signal path that a signal shall not take in the topology (ForbiddenSignalPath)

Class	ForbiddenSignalPath			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The ForbiddenSignalPath describes the physical channels which an element shall not take in the topology. Such a signal path can be a constraint for the communication matrix, because such a path has an effect on the frame generation and the frame path.			
Base	ARObject, SignalPathConstraint			
Attribute	Type	Mult.	Kind	Note
operation	SwcToSwcOperationArguments	*	aggr	Reference to the operation arguments of one operation which shall not take the predefined way in the topology.
physical Channel	PhysicalChannel	1..*	ref	The SwcToSwcSignal shall not be transmitted on one of these physical channels.
signal	SwcToSwcSignal	*	aggr	The data element which shall not take the predefined way in the topology.

Table 5.31: ForbiddenSignalPath

5.2.2.3 PermissibleSignalPath

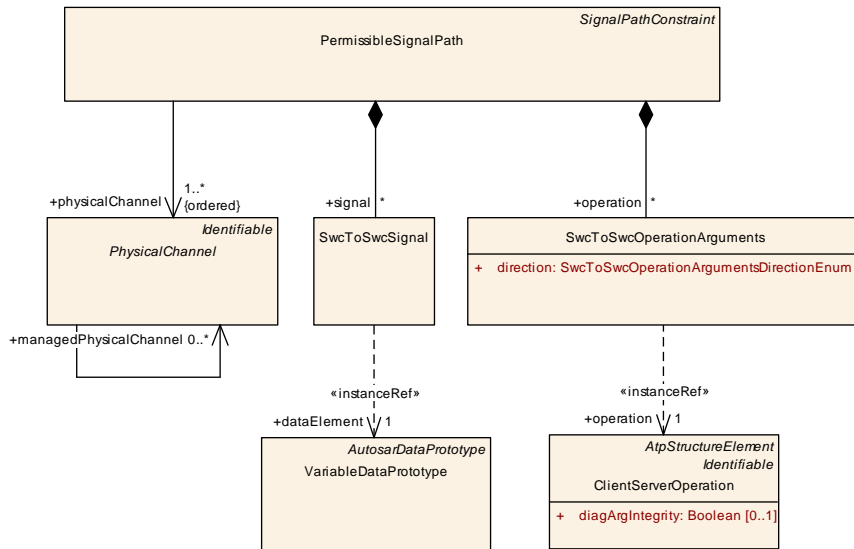


Figure 5.20: Description of the signal path that a signal shall take in the topology (PermissibleSignalPath)

Class	PermissibleSignalPath			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	<p>The PermissibleSignalPath describes the way a data element shall take in the topology. The path is described by ordered references to PhysicalChannels.</p> <p>If more than one PermissibleSignalPath is defined for the same signal/operation attributes, any of them can be chosen. Such a signal path can be a constraint for the communication matrix . This path describes that one data element should take path A (e.g. 1. CAN channel, 2. LIN channel) and not path B (1. CAN channel, FlexRay channel A).</p> <p>This has an effect on the frame generation and the frame path.</p>			
Base	ARObject, SignalPathConstraint			
Attribute	Type	Mult.	Kind	Note
operation	SwcToSwcOperationArguments	*	aggr	The arguments of an operation that can take the predefined way in the topology.
physical Channel (ordered)	PhysicalChannel	1..*	ref	The SwcToSwcSignal can be transmitted on one of these physical channels.
signal	SwcToSwcSignal	*	aggr	The data element which can take the predefined way in the topology.

Table 5.32: PermissibleSignalPath

5.2.2.4 SeparateSignalPath

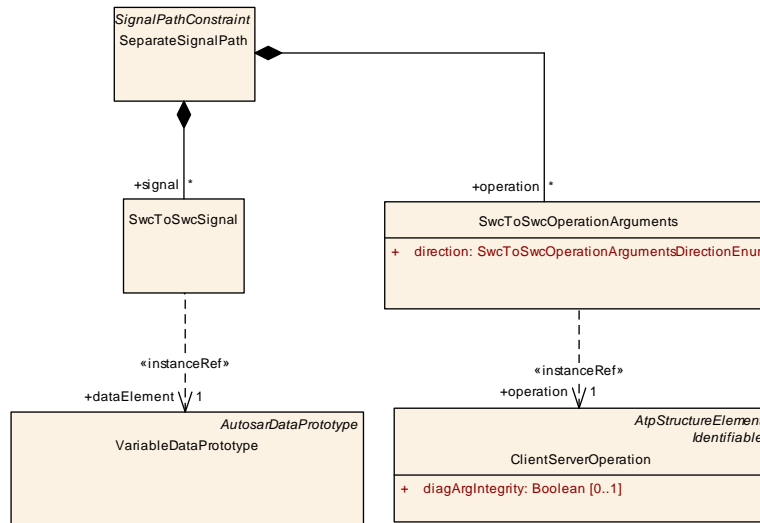


Figure 5.21: Description of signals that shall not take the same way in the topology (SeparateSignalPath)

Class	SeparateSignalPath			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SeparateSignalPath describes that two SwcToSwcSignals and/or SwcToSwcOperationArguments shall not take the same way (Signal Path) in the topology (e.g. Redundancy). This means that the signals are not allowed to share even a single physical channel in their path.			
Base	ARObject, SignalPathConstraint			
Attribute	Type	Mult.	Kind	Note
operation	SwcToSwcOperationArguments	*	aggr	The SwcToSwcOperationArguments that shall not take the same way (Signal Path) in the topology.
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that shall not take the same way (Signal Path) in the topology.

Table 5.33: SeparateSignalPath

5.3 RTE and basic software resource estimations

Important constraints for system partitioning are the available resources on the ECUs in the system. For SW components, the resource estimations can be stated in SW component descriptions. It is however not only SW components that require resources. AUTOSAR RTE and basic software running on the ECU have resource needs as well.

The realization of the RTE and the kind of basic software to be run on a certain ECU depend on the implicit and explicit usage of all basic software by the software components. The software components need to communicate internally and with software components on other ECUs. Furthermore, they have different needs with respect to scheduling. This results in implicit use of e.g. communication and operating system software. In addition, the software components make explicit use of basic software when they e.g. utilize system services (e.g. diagnostics) and access sensors/actuators via the I/O abstraction layer or the Complex Driver abstraction layer. Thus, the resource consumption of the RTE and the basic software depend on the SW Components mapped to the ECU, since this determines the exact configuration of the RTE and the basic software.

[TPS_SYST_01126] Resource Consumption for RTE and basic software [The resource consumption for RTE and basic software may be specified using class `EcuResourceEstimation`. Each estimation is performed for a specific ECU and for a specific set of SW mapped to that ECU (reference from `EcuResourceEstimation` to `EcuInstance` and `SwcToEcuMapping`).] (*RS_SYST_00002*)

Different resource estimations for a specific ECU, but with different mappings may exist, e.g. for different variants of the system, or to show the difference of resource needs for different mappings. The `EcuResourceEstimation` aggregates the meta-class `ResourceConsumption` from the `GenericStructure` package each for RTE and basic software, which specifies stack and heap usage and execution time.

`ExecutionTime` and `StackUsage` are used to provide information on the implementation specific resource usage of the `ExecutableEntity` defined in the `InternalBehavior` of SW-Component respectively in the `BswInternalBehavior` of BSW Module. `MemorySection` documents the resources needed to load the object file containing the implementation on the ECU. `HeapUsage` describes the dynamic memory usage of the software.

Figure 5.22 shows the meta-model for resource estimations for RTE and basic SW.

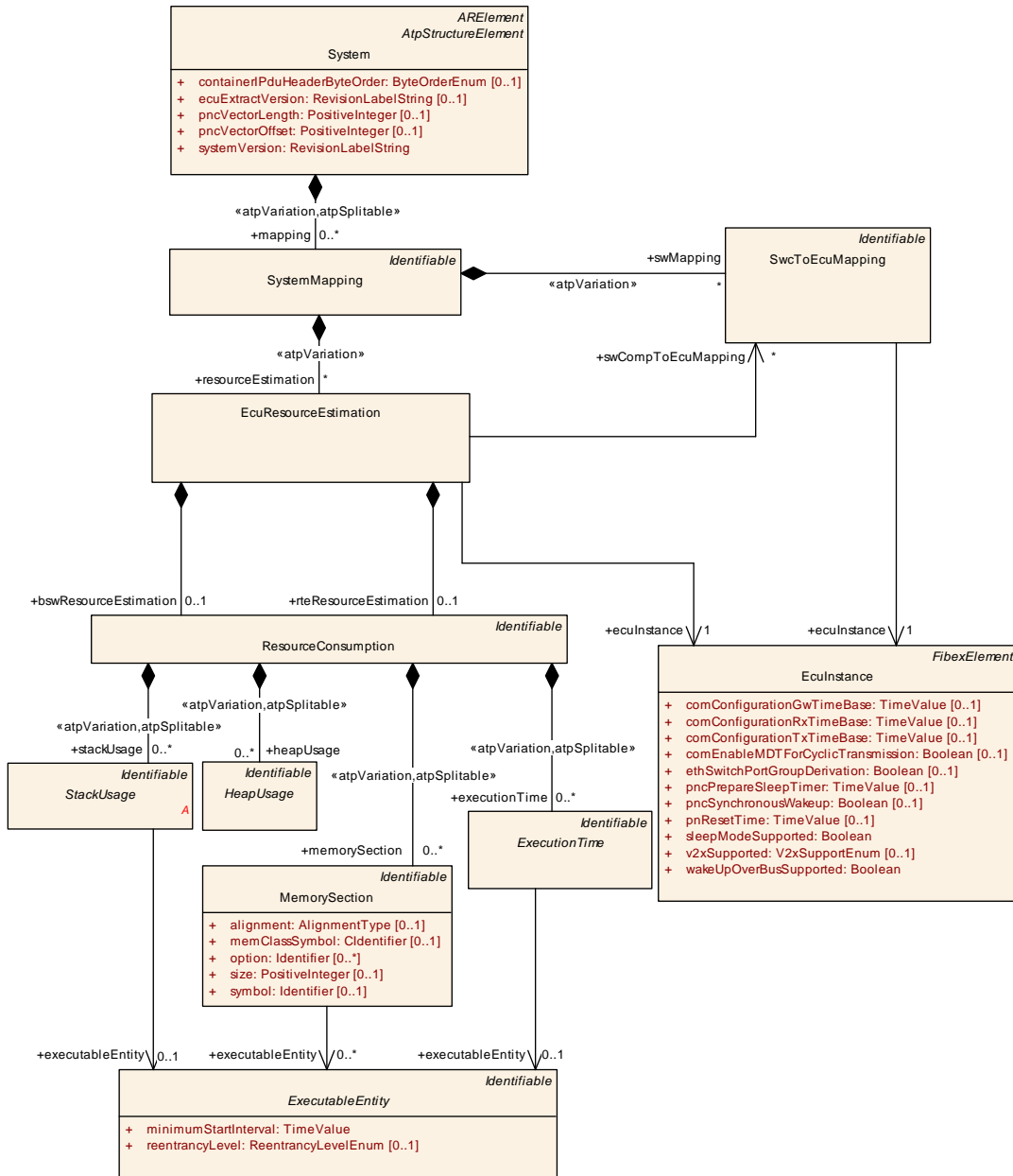


Figure 5.22: ECU resource estimations (ResourceEstimation)

[constr_3005] valid **EcuResourceEstimation** [The same **EcuInstance** shall be referenced directly from the **EcuResourceEstimation** and from the **SwcToEcuMapping**:

`EcuResourceEstimation.swCompToEcuMapping.ecuInstance == EcuResourceEstimation.ecuInstance]` ()

Class	EcuResourceEstimation			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
bswResourceEstimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the basic software.
ecuInstance	EcuInstance	1	ref	Reference to the ECU this estimation is done for.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ecu resource estimation Tags: xml.sequenceOffset=-10
rteResourceEstimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the run time environment.
swCompToEcuMapping	SwcToEcuMapping	*	ref	References to SwcToEcuMappings that have been taken into account for the resource estimations. This way it is possible to define different EcuResourceEstimations with different mappings, e.g. before and after mapping an additional SW component.

Table 5.34: EcuResourceEstimation

Class	ResourceConsumption			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption			
Note	Description of consumed resources by one implementation of a software.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
accessCountSet	AccessCountSet	*	aggr	Set of access count values Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=accessCountSet, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
executionTime	ExecutionTime	*	aggr	Collection of the execution time descriptions for this implementation. The aggregation of executionTime is subject to variability with the purpose to support the conditional existence of runnable entities. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
heapUsage	HeapUsage	*	aggr	Collection of the heap memory allocated by this implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
memorySection	MemorySection	*	aggr	An abstract memory section required by this Implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime





Class	ResourceConsumption			
sectionName Prefix	SectionNamePrefix	*	aggr	A prefix to be used for the memory section symbol in the code. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
stackUsage	StackUsage	*	aggr	Collection of the stack memory usage for each runnable entity of this implementation. The aggregation of Stack Usage is subject to variability with the purpose to support the conditional existence of runnable entities. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 5.35: ResourceConsumption

The element [ResourceConsumption](#) and the subelements [HeapUsage](#), [Memory-Section](#), [StackUsage](#) and [ExecutionTime](#) are described in more detail in the BSW Module Description [18].

5.4 Partial Networking

The AUTOSAR BSW stack supports power saving during vehicle operation time with the partial networking mechanism. This mechanism allows to shut down and startup the bus communication interfaces of groups of ECUs (Partial Network Cluster) during normal bus communication.

On the VFB Level Partial Networks are represented by Virtual Function Clusters and are described with [PortGroups](#). The Virtual Function Cluster groups the communication necessary to realize one or more vehicle functions that can become activated/deactivated during normal vehicle operation. Virtual Function Clusters are described in more detail in [5]. The Virtual Function Clusters are mapped onto Partial Network Clusters.

[TPS_SYST_01133] Partial Network Clusters [Partial Network Clusters are realized with [ISignalIPduGroups](#) using [PncMapping](#).] ([RS_SYST_00042](#))

Each [PncMapping](#) has the ability to define relations to [ISignalIPduGroups](#), [PdurIPduGroups](#), and [PhysicalChannels](#). These relations are used to describe the impact of a PNC on the respective networking artifacts. The realization of those relationships is typically to be implemented in the Basic Software Mode Manager (BswM).

- [PncMapping.pncGroup](#) (referring to [ISignalIPduGroup](#)) is used to define which Com [ISignalIPduGroups](#) shall be enabled when this PNC is active. The use-case is to enable/disable Com [ISignalIPdus](#), especially periodic behavior like cyclic sending and time-out monitoring.

- `PncMapping.pncPdurGroup` (referring to `PdurIPduGroup`) is used to define which `PdurIPduGroups` shall be enabled when this PNC is active. The use-case is to enable/disable Pdus which typically are not passing the Com module (e.g. Pdus going through LdCom, Diagnostic IPdus, Gatewayed IPdus).
- `PncMapping.physicalChannel` (referring to `PhysicalChannel`) is used to define which `PhysicalChannels` shall be enabled when this PNC is active. The use-case is to enable/disable the whole `PhysicalChannel` and allows to also cover Pdus which neither pass the Com module nor the Pdu Router module (e.g. NM).

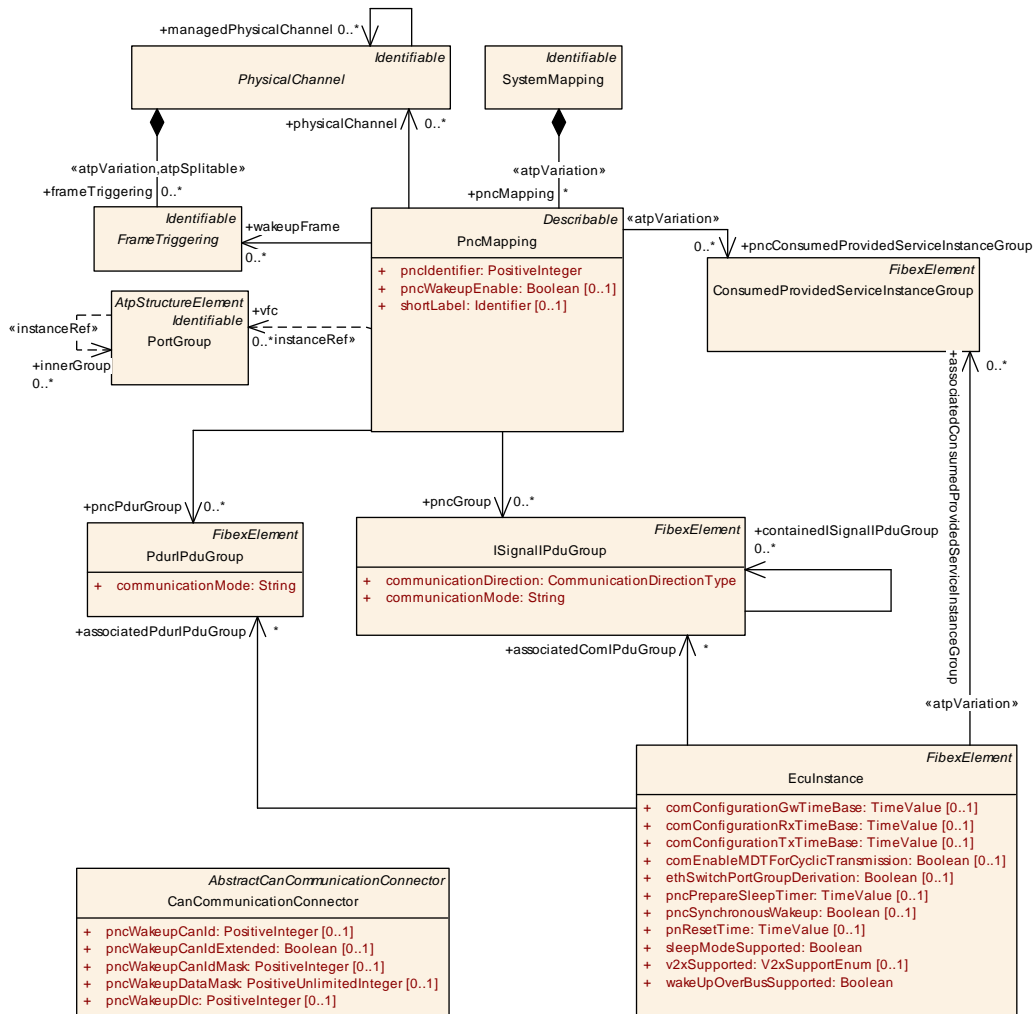


Figure 5.23: Mapping of Virtual Function Clusters onto Partial Network Clusters

Class		PncMapping		
Package		M2::AUTOSARTemplates::SystemTemplate::PncMapping		
Note		Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.		
Base		ARObject, Describable		
Attribute	Type	Mult.	Kind	Note
ident	PncMappingIdent	0..1	aggr	This adds the ability to become referable to PncMapping.
physical Channel	PhysicalChannel	*	ref	This reference maps the partial network to a communication channel.
pncConsumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	ConsumedProvidedServiceInstanceGroup used in a Partial Network Cluster. This reference is optional, since this could be used for starting and stopping Consumed ProvidedServiceInstanceGroup according the requested partial network, but is not necessarily needed. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
pncGroup	ISignalIPduGroup	*	ref	IPduGroup participating in a Partial Network Cluster. This reference is optional in case an ecu extract has only indirect pnc access, i.e. ecu is not directly connected to a network which supports partial network.
pncIdentifier	PositiveInteger	1	attr	Identifier of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu.
pncPdurGroup	PdurIPduGroup	*	ref	This reference maps the Partial Network Cluster to a set of PdurIPduGroups.
pncWakeup Enable	Boolean	0..1	attr	If this parameter is available and set to true then this PNC will be woken up as soon as a channel wakeup occurs on a channel where this PNC is assigned to. This is ensured by adding this PNC to the corresponding channel wakeup sources during upstream mapping.
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the PncMapping. It shall be unique in the System scope.
vfc	PortGroup	*	iref	Virtual Function Cluster to be mapped onto a Partial Network Cluster. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems.
wakeupFrame	FrameTriggering	*	ref	Reference to collection of FrameTriggerings that are used for the wakeup of this PNC (Application Frames or Nm Frames can be used). This reference is only valid if this EcuExtract represents an ECU which has direct PNC access, i.e. ECU is directly connected to a network which supports partial network.

Table 5.36: PncMapping

[constr_3039] pncIdentifier range [The `pncIdentifier` value shall be in the range of 8..63.]()

[constr_3198] Uniqueness of PncMapping.shortLabel [If the optional `shortLabel` attribute is used it shall be unique in the `System` scope.]()

The runtime information that is used to coordinate the request/release information of all partial networks is called `pncVector`. The size and position of the `pncVector` inside the network management user data (`NmPdu.iSignalToIPduMapping`) is globally defined in the `System` class in chapter 1.6.

In the system description the `NmPdus` are described based on the actual network interaction (i.e. an ECU sends one `NmPdu` per network and receives a set of `NmPdus`).

`NmPdus` that define the existence of NM user data via the existence of the attribute `iSignalToIPduMapping` shall be referenced by corresponding `PduTriggerings` where the attribute `iPduPort` exists accordingly. This is also reflected by [TPS_SYST_01057].

`NmPdus` that define the existence of NM user data via the definition of `NmPdu.nm-DataInformation` shall not be referenced by `PduTriggerings` because neither `Com` nor `PduR` are involved in the transmission (which lets the `Nm` module talk to the bus interface directly).

The `NmPdus` contributing to partial networking also have the `Nm` user data layout specified to contain the `pncVector` as an `UINT8_N` signal.

Those `Pdus` which are used to perform the ECU internal communication between the basic software modules (like *EIRA*, *ERA*) are not described in the system description and need to be introduced to the ECU Configuration.

[constr_3040] Restriction of `pncIdentifier` values [The `pncIdentifier` value shall be within the range described by `pncVectorOffset` and `pncVectorLength`.]
()

[constr_3041] `pncVectorOffset` range [The `pncVectorOffset` value shall be in the range of 1..7.]()

[constr_3042] `pncVectorLength` range [The `pncVectorLength` value shall be in the range of 1..7.]()

Please note that a `pncVector` is configured as `UINT8_N` signal in AUTOSAR `Com`.

[constr_3146] Partial Networking timing constraint [For Partial Networking the following timing constraints shall be ensured:

- CAN / Ethernet: $(pnResetTime + pncPrepareSleepTimer) < nmNetworkTimeout$
- FlexRay: $(pnResetTime + pncPrepareSleepTimer) < nmReadySleepTime$

]()

[TPS_SYST_02145] Default behavior for not defined `nmPncParticipation` [When `NmCluster.nmPncParticipation` is set to *true* or is not defined this `NmCluster` shall contribute to the partial network mechanism.]()

[constr_3323] Relation between `NmCluster.nmPncParticipation` and `PncMapping.pncGroup` [If a `PncMapping` references an `ISignalIPduGroup` in role `pncGroup` which in turn

- contains (either directly or via one of its subordinate `ISignalIPduGroups` referenced in role `containedISignalIPduGroup`) `ISignalIPdus` that are referenced by a `PduTriggering` in role `iPdu` which in turn

- is composed by a `PhysicalChannel` in role `pduTriggering` which in turn
- is composed by `CommunicationCluster` in role `physicalChannel` which in turn
- is referenced by an `NmCluster` in role `communicationCluster`,

then this `NmCluster` shall have its `nmPncParticipation` attribute set to TRUE unless the `PhysicalChannel` is referenced in the role `managedPhysicalChannel`.]()

[constr_3484] PncMapping that refers a managedPhysicalChannel shall also refer the managing PhysicalChannel [If a `PncMapping` refers to a `PhysicalChannel` (either directly in the role `physicalChannel` or indirectly by referencing an `ISignalIPduGroup` in the role `pncGroup`) and this `PhysicalChannel` is referenced in the role `managedPhysicalChannel`, then the according managing `PhysicalChannel` (the source of the `managedPhysicalChannel` reference) shall also be referenced by the `PncMapping` (either directly in the role `physicalChannel` or indirectly by referencing an `ISignalIPduGroup` in the role `pncGroup`).]()

Note that **[constr_3484]** ensures that the managing `PhysicalChannel` is part of the same PNC as the `managedPhysicalChannels`.

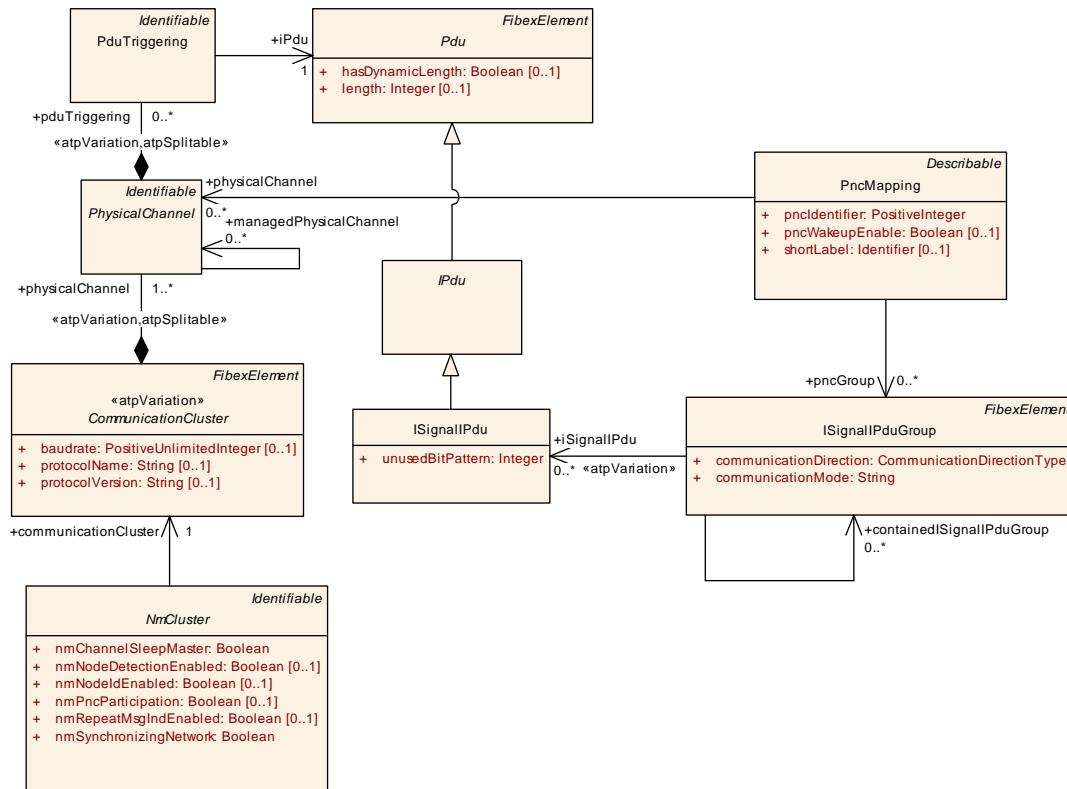


Figure 5.24: Relation between `NmCluster.nmPncParticipation` and `PncMapping.pncGroup`

[TPS_SYST_02146] Explicit definition of `pncVector` at `NmPdu` [If there is an `ISignalToIPduMapping` aggregated by `NmPdu` that fully matches the interval defined by `pncVectorOffset` and `pncVectorLength` then the corresponding `ISignal` represents the `pncVector`.]()

[TPS_SYST_02147] Implicit definition of `pncVector` at `NmPdu` [If there is no `ISignalToIPduMapping` aggregated by `NmPdu` that fully matches the interval defined by `pncVectorOffset` and `pncVectorLength` and the respective `NmCluster.nmPncParticipation` has the value *true* or is not defined then the `ComSignal` in the COM configuration shall be derived locally in the Ecu Configuration according to the following rules: `ComBitPosition` and `ComSignalLength` shall be derived from `pncVectorOffset` and `pncVectorLength` respectively. Since `ComSignalType` is `UINT8_N`, `ComSignalEndianness` shall be `OPAQUE` (see `SWS_Com_00553`). `ComTransferProperty` shall be set to `PENDING` and `ComSignalInitValue` shall be set to 0.]()

Attributes used to configure the Partial Network Wakeup of one specific Ecu are described in chapter 3.3.1.4.

5.4.1 Partial Networking and managed Ethernet switch

On switched Ethernet networks it is possible to let the Ethernet switch be managed by an AUTOSAR Ecu. In this case the configuration and the behavior of the `CouplingElement` with `couplingType=switch` are controlled by the management Ecu.

For the usage of Partial Networking on switched Ethernet networks with managed `CouplingElements` an additional application arises: Depending on the requested Partial Networks it shall be possible to switch off Ethernet switch ports which are not involved in any communication currently active.

In order to describe the relationships between Partial Networks and Ethernet switch ports an optional reference from `CouplingPort` to a `PncMappingIdent` in the role `pncMapping` is defined.

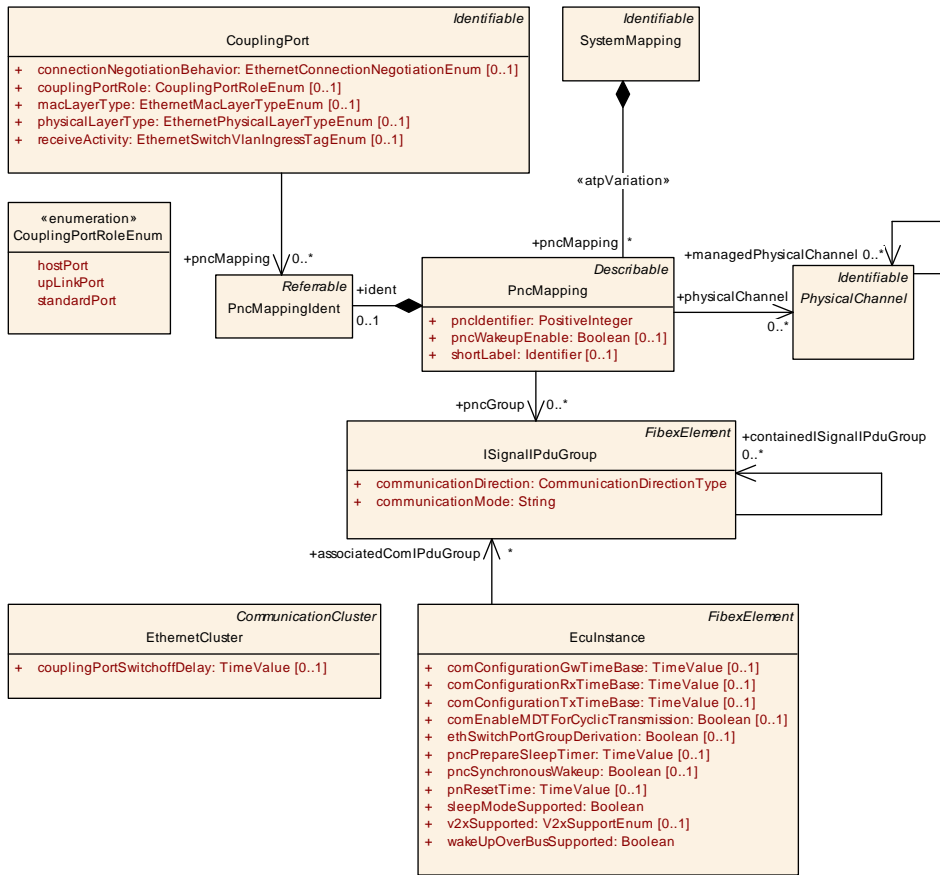


Figure 5.25: Partial Networking and managed Ethernet switch

The example in figure 5.26 illustrates the setup of an management ECU which manages 2 Ethernet switches.

The port 1 of Switch 1 is a `hostPort`.

The port 5 of Switch 1 and port 1 of Switch 2 are `upLinkPorts`.

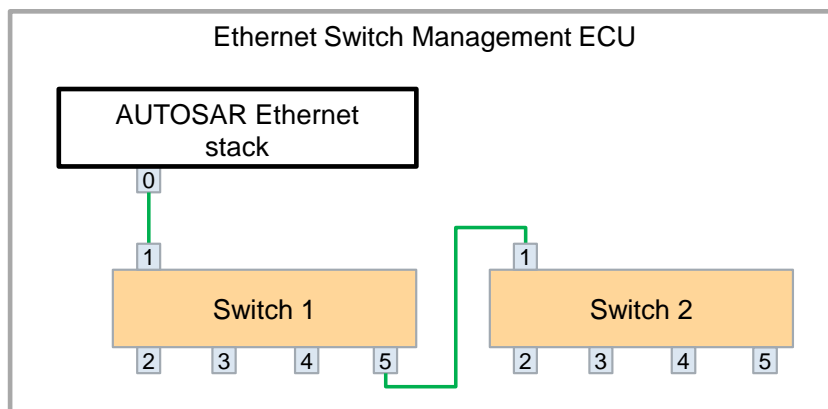


Figure 5.26: Example of managed Ethernet switches

[constr_3523] CouplingPort and PncMapping in the scope of an Ethernet-PhysicalChannel | If

- a `CouplingPort` referring to an `EthernetPhysicalChannel` – via a `VlanMembership` – references at least one `PncMapping`
- and that `PncMapping` contains PDUs – via the assignment of `PncMapping.pncGroup` – that are transported on this `EthernetPhysicalChannel`

then every `CouplingPort` referring to that `EthernetPhysicalChannel` shall reference at least one `PncMapping` as well.]()

If a `CouplingPort` referring to an `EthernetPhysicalChannel` – via a `VlanMembership` – references no `PncMapping` then any other `CouplingPort` referring to that `EthernetPhysicalChannel` is allowed to either reference `PncMappings` or not.

[TPS_SYST_03018] Aggregation of PNCs at the `hostPort` [A `CouplingPort` with `couplingPortRole` set to `hostPort` shall reference all `PncMappings` that are referenced by any `CouplingPorts` of the same `CouplingElement` and all `CouplingElements` connected to this `CouplingElement`.]()

[constr_3524] Definition of `couplingPortRole` on `CouplingPort` for managed `CouplingElement` [A managed `CouplingElement` shall have either

- at most one `CouplingPort` with `couplingPortRole` set to `hostPort` or
- at least one `CouplingPort` with `couplingPortRole` set to `upLinkPort`.

]()

[constr_3525] Connection of `CouplingPort` with `couplingPortRole` set to `upLinkPort` [A `CouplingPort` with `couplingPortRole` set to `upLinkPort` shall be connected to exactly one other `CouplingPort` with `couplingPortRole` set to `upLinkPort`.]()

[TPS_SYST_03020] Default value for `CouplingPort.couplingPortRole` if not defined [If no value for the attribute `CouplingPort.couplingPortRole` is defined then `standardPort` shall be assumed.]()

[TPS_SYST_03019] Modeling of `CouplingPorts` for managed `CouplingElement` [Only `CouplingPorts` that participate in the communication of a managed `CouplingElement` shall be modeled in the System Description.]()

All other ports of an Ethernet switch are not modeled. The expected behavior of unmodeled Ethernet switch ports on runtime:

1. the Ethernet switch driver switches off this Ethernet switch ports during its initialization
2. unmodeled Ethernet switch ports shall never be switched on.

5.4.2 Partial Network Gateway

As Partial Networks are spread over multiple networks their state has to be synchronized by gateway ECUs connected to a Partial Network on more than one network. Therefore a gateway mechanism is available where per network it can be configured if the network takes part in Partial network gateway algorithm or not. When more than one gateway is connected to one network then only one gateway shall be configured as main gateway.

[constr_5093] pncGatewayType and PhysicalChannel [When multiple `CommunicationConnectors` with `pncGatewayType` set to a value other than `none` are referenced by the same `PhysicalChannel` then only up to one `CommunicationConnector` shall have the `pncGatewayType` set to `active`.]()

An ECU having a Partial Network connected to more than one network may not necessarily coordinate this Partial Network. There can be uses-cases where such an ECU is connected to multiple networks where this Partial Network is relevant but all of these networks are handled by the same gateway or by various gateways connected on another network. In this case the ECU does not coordinate the Partial Network.

[constr_5094] pncGatewayType and ECU [When an ECU is connected to more than one `PhysicalChannel` and has a relation to a Partial Network then all `CommunicationConnectors` of this ECU where this Partial Network is related to shall have the `pncGatewayType` value either set to `none` or to a value different than `none` (i.e. `active` or `passive`).]()

Hint: This constraint ensures that an ECU either coordinates a Partial Network on all of its connected channels or on none.

5.5 Com Management Mapping

The AUTOSAR BSW stack supports the configuration of the ComM module which encapsulates the control of the underlying communication services. The ComM module collects the bus communication access requests from communication requestors and coordinates the bus communication access requests. In order to utilize the communication requests from application software the Software Component Template supports the definition of `PortGroups` with the category `MODE_MANAGEMENT`. In this section it is described how `PortGroups` with the category `MODE_MANAGEMENT` are mapped to communication channels.

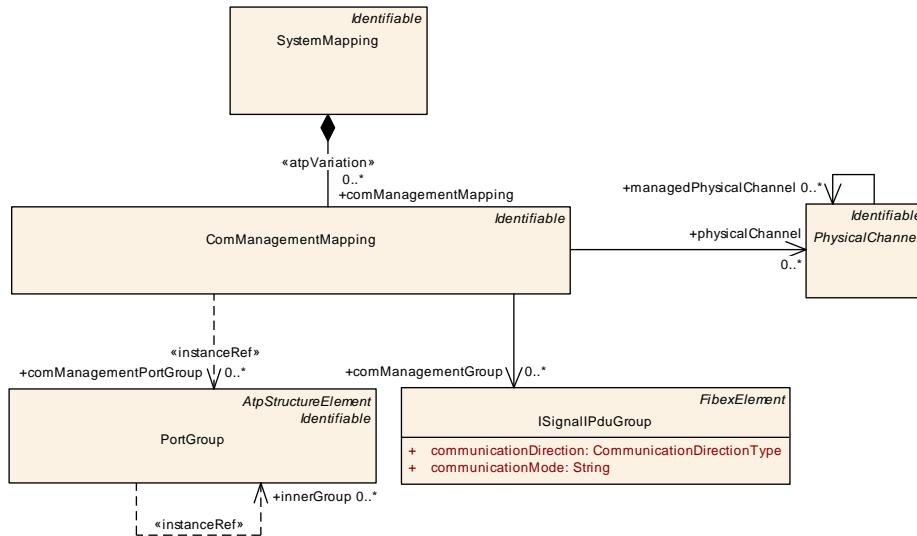


Figure 5.27: Mapping between PortGroups and communication channels

Class	ComManagementMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	Describes a mapping between one or several Mode Management PortGroups and communication channels.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
com Management Group	ISignalPduGroup	*	ref	IPduGroup participating in a Mode Management Port Group.
com Management PortGroup	PortGroup	*	iref	Mode Management PortGroup to be mapped onto a communication channel. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems.
physical Channel	PhysicalChannel	*	ref	This reference maps the Mode Management PortGroup partial network to communication channels.

Table 5.37: ComManagementMapping

6 Communication

This chapter describes all topics that deal with constraints or configurations that describe the information exchange between the ECUs. The description of communication matrices in the System Template is based on the description in ASAM FIBEX [9]. Because of the requirements of AUTOSAR some extensions were made to the original FIBEX model.

The main elements to describe communication in the System Template are [System-Signals](#), [ISignals](#), [Pdus](#) and [Frames](#), as it can be seen on Figure 6.1.

[Frames](#) can be defined independently of communication clusters. On the communication channel the [Frame](#) is represented by the referencing [FrameTriggering](#).

A [Frame](#) has a payload section of a certain length in bytes, which contains an arbitrary number of non-overlapping [Pdus](#). In AUTOSAR only FlexRay supports the packing and unpacking of multiple [Pdus](#) into/out of one FlexRay [Frame](#). The AUTOSAR CanIf and LinIf are not capable of packing multiple [Pdus](#) into one [Frame](#).

[constr_3036] Pdus in CAN and LIN Frames [CAN Frames and LIN Frames shall only contain one [Pdu](#).]()

Note that via the [ContainerIPdu](#) it is possible to transport several [IPdus](#) in one [ContainerIPdu](#) in order to support CAN FD.

A [Pdu](#) (Protocol Data Unit) is the information delivered through a network layer. For the network to understand which layer is being discussed, a single-letter prefix is added to the PDU.

- [IPdu](#) - Interaction Layer Protocol Data Unit (assembled and disassembled in Com). In the case of external communication the Interaction Layer packs one or more signals into assigned [IPdus](#) and passes them to the underlying layer for transfer between nodes in a network.
- [NPdu](#) - Network Layer Protocol Data Unit (assembled and disassembled in a Transport Protocol module). The TP module's main purpose is the segmentation and reassembly of [IPdus](#) that do not fit in one of the assigned [NPdus](#).
- [LPdu](#) - Data Link Layer Protocol Data Unit (assembled and disassembled in AUTOSAR Hardware Abstraction layer). The element [Frame](#) in the System Template represents the Autosar Layered Architectures [LSdu](#). [Sdu](#) is the abbreviation of "Service Data Unit". The Data Link Layers [LPdu](#) contains the [LSdu](#) and [PCI](#) (Protocol Control Information). The [LPdu](#) is not described in the System Template.

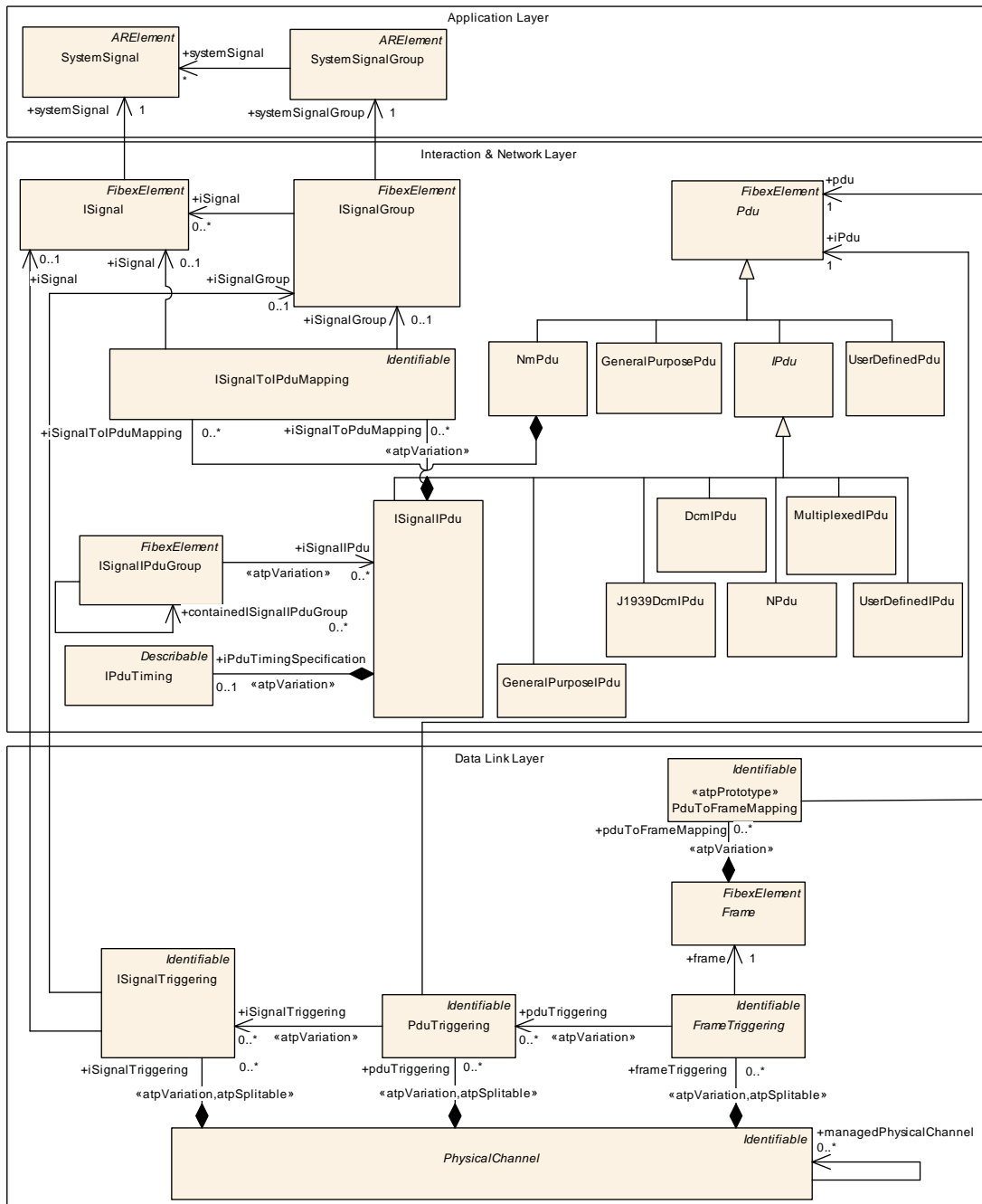


Figure 6.1: Communication Overview (FibexCore: Communication)

In case no multiplexing is performed the *IPdu*s of Com that fit into one frame can be passed directly via the PDU Router to the communication interfaces.

[constr_3037] maximum Frame frameLength for CAN and LIN [For CAN and LIN the maximum *frameLength* is 8 bytes and 64 bytes in case of CAN FD.]()

[constr_3038] maximum Frame frameLength for FlexRay [For FlexRay the maximum *frameLength* is 254 bytes.]()

[TPS_SYST_01048] Handling of large **IPdus** [Large **IPdus** that are too long to fit into one **Frame** of the respective subclass of **CommunicationCluster** shall be routed via a Transport Protocol to the communication interfaces.]()

For example an **IPdu** with the length of 10 bytes needs to be routed via a Transport Protocol on CAN but on FlexRay this is not required.

The Transport Protocols are described in more detail in chapter 6.8.

If multiplexing is performed an **IPdu** is routed between the **IPdu Multiplexer** and the **Interface Layer** or **Transport Layer**. To distinguish these two different cases two specializations **ISignalIPdu** and **MultiplexedIPdu** are introduced. A **ISignalIPdu** represents an **IPdu** handled by AUTOSAR Com. The AUTOSAR **IPduM** is responsible to combine Com **ISignalIPdus** to **MultiplexedIPdus**. On receiver-side the **IPduM** is responsible to interpret the content of **MultiplexedIPdus** and provide Com separated **ISignalIPdus** by taking into account the value of the selector field. The **IPdu Multiplexer** is described in more detail in chapter 6.5.

AUTOSAR Com provides the possibility to define **Transmission Modes** for each Com **ISignalIPdu**. For this reason the **ISignalIPdu** aggregates the **IPduTiming**. The **Transmission Modes** are described in more detail in chapter 6.4.

6.1 Triggerings and Ports

The elements **FrameTriggering**, **PduTriggering** and **ISignalTriggering** are describing the usage of **Frames**, **IPdus** and **ISignals** on a **PhysicalChannel**.

A **FrameTriggering** need to fulfill requirements for contained **Pdus** that are defined by the corresponding **PduTriggerings**. And the **PduTriggering** need to fulfill requirements for contained **ISignals** that are defined by the corresponding **ISignalTriggerings**. The references between the **Triggering** elements can be used to describe these relationships. More details can be found in class tables of **FrameTriggering**, **PduTriggering** and **ISignalTriggering**.

In AUTOSAR the timing of bus messages can be controlled by send requests of the **Application layer** in combination with the **Com Transmission Modes** and **Transfer Properties** (esp. CAN). On the other hand it can be controlled by the **FlexRay** or **LIN Interface**. In this case the **Bus Interface** only requests **IPdus** that have to be provided by **Com**.

In the **System Template** the **Com controlled timing** is described with the aggregation between the **ISignalIPdu** and the **IPduTiming**. The **LIN** and **FlexRay Scheduling Tables** are described in the **FrameTriggering**.

Timing requirements for **FlexRay**, **TTCAN** and **LIN Pdus** can be specified with the **Timing Extension model**. More details are described in chapter 1.7.3.

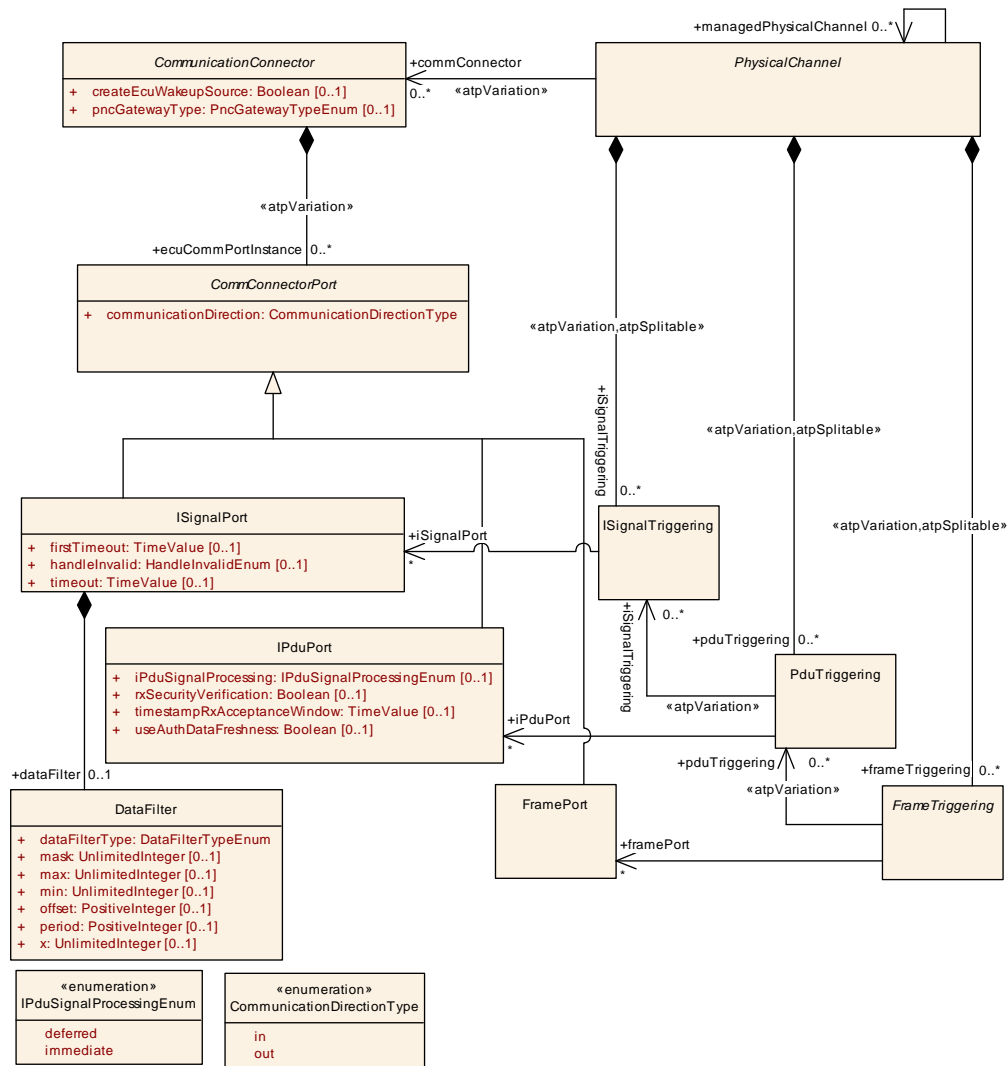


Figure 6.2: Communication Matrix (FibexCore: CommunicationMatrix)

Figure 6.2 shows the relationship between the `CommConnectorPort` and the `FrameTriggering`, `PduTriggering` and `ISignalTriggering`. This relationship allows to specify explicitly which `Frames`, `Pdus`, `ISignals` are received/sent by the connected ECU on the connected channel.

[constr_3243] FrameTriggering.pduTriggering condition [A `FrameTriggering` shall reference a `PduTriggering` if the `PduTriggering` references a `Pdu` that is referenced by a `PduToFrameMapping` which in turn is aggregated by the `Frame` that is referenced by that `FrameTriggering`.]()

[constr_3250] PduTriggering.iSignalTriggering condition [A `PduTriggering` shall reference an `ISignalTriggering` if the `ISignalTriggering` references an `ISignal` or an `ISignalGroup` that is referenced by an `ISignalToIPduMapping` which in turn is aggregated by the `Pdu` that is referenced by that `PduTriggering`.]()

[TPS_SYST_02102] `FrameTriggering.pduTriggering` references that shall be ignored [References from `FrameTriggering` to `PduTriggering` which are not covered by `[constr_3243]` shall be ignored.]()

As a consequence of `[constr_3243]` the following implications can be derived:

- The `PduTriggering` of the `ContainerIPdu` is referenced from the `FrameTriggering` but the `PduTriggerings` of the contained `IPdus` are not referenced from the `FrameTriggering`.
- The `PduTriggering` of the `MultiplexedIPdu` is referenced from the `FrameTriggering` but the `PduTriggerings` of the multiplexed Part `Pdus` are not referenced from the `FrameTriggering`.

[TPS_SYST_02104] Triggerings on `PhysicalChannel` [The following modeling creates a "membership" of `ISignals`, `ISignalGroups`, `Pdus`, and `Frames` in a given `PhysicalChannel`:

- `PhysicalChannel` aggregates
 - `ISignalTriggering` that in turn references `ISignal` in the role `iSignal`
 - `ISignalTriggering` that in turn references an `ISignalGroup` in the role `iSignalGroup` (`[TPS_SYST_02105]` applies).
- `PhysicalChannel` aggregates `PduTriggering` that in turn references a `Pdu` in the role `iPdu`.
- `PhysicalChannel` aggregates `FrameTriggering` that in turn references a `Frame` in the role `frame`.

]()

[TPS_SYST_02105] `ISignalGroup` and `ISignal` referenced from `ISignalTriggering` [Either an `ISignalGroup` and all `ISignals` referenced from the `ISignalGroup` are also referenced from `ISignalTriggerings` aggregated at the same `PhysicalChannel` or neither the `ISignalGroup` nor any of the `ISignals` referenced by the `ISignalGroup` shall be referenced from `ISignalTriggerings`.]
()

[TPS_SYST_01142] Rules for the creation of references to Ports (`ecuCommPortInstance`) with `communicationDirection out` on sending Ecu [

- Application sends `ISignal` or `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup` members that are sent by the Application.

- `PduTriggering` reference to `IPduPort` shall be created
- `FrameTriggering` reference to `FramePort` shall be created
- COM Signal Gateway
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` for a subset of `ISignals` inside the `ISignalGroup` shall be created (in case not all members of the `ISignalGroup` participate in the target Signal Gateway relation).
 - `PduTriggering` reference to `IPduPort` shall be created
 - `FrameTriggering` reference to `FramePort` shall be created
- `ISignal` or `ISignalGroup` is mapped to `ISignalIPdu` but NOT sent by Application or Signal Gateway
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for any `ISignal` that is part of an `ISignalGroup`
 - `PduTriggering` reference to `IPduPort` shall be created
 - `FrameTriggering` reference to `FramePort` shall be created
- Neither `ISignal`, `ISignalGroup`, `Pdu`, nor `Frame` sent by the ECU
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for any `ISignal` that is part of an `ISignalGroup`
 - No `PduTriggering` reference to `IPduPort` shall be created
 - No `FrameTriggering` reference to `FramePort` shall be created

]0

Please note that it is possible to configure a signal that is transmitted by an application and also routed by a SignalGateway. At runtime it has to be ensured that only one path is active at a particular point in time (to avoid race conditions in COM Stack).

[TPS_SYST_02106] Rules for the creation of references to Ports (`ecuComm-PortInstance`) with `communicationDirection` in on receiving Ecu [

- Application receives `ISignal` or `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup` members that are received by the Application.
 - `PduTriggering` reference to `IPduPort` shall be created
 - `FrameTriggering` reference to `FramePort` shall be created
- COM Signal Gateway
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` shall be created for the `ISignalGroup`
 - Reference from `ISignalTriggering` to `ISignalPort` for a subset of `ISignals` inside the `ISignalGroup` shall be created (in case not all members of the `ISignalGroup` participate in the source Signal Gateway relation).
 - `PduTriggering` reference to `IPduPort` shall be created
 - `FrameTriggering` reference to `FramePort` shall be created
- `ISignal` or `ISignalGroup` is mapped to `ISignalIPdu` but NOT received by Application or Signal Gateway
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for any `ISignal` that is part of an `ISignalGroup`
 - `PduTriggering` reference to `IPduPort` shall be created
 - `FrameTriggering` reference to `FramePort` shall be created

- Neither `ISignal`, `ISignalGroup`, `Pdu`, nor `Frame` received by the ECU
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignal` that is not part of an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for an `ISignalGroup`
 - No `ISignalTriggering` reference to `ISignalPort` shall be created for any `ISignal` that is part of an `ISignalGroup`
 - No `PduTriggering` reference to `IPduPort` shall be created
 - No `FrameTriggering` reference to `FramePort` shall be created

]()

[constr_3252] `ISignalTriggering.iSignalPort` reference condition [An `ISignalTriggering` shall only reference an `ISignalPort` if the `CommunicationConnector` aggregating that `ISignalPort` is referenced by the `PhysicalChannel` which in turn aggregates that `ISignalTriggering`.]()

[constr_3253] `PduTriggering.iPduPort` reference condition [A `PduTriggering` shall only reference an `IPduPort` if the `CommunicationConnector` aggregating that `IPduPort` is referenced by the `PhysicalChannel` which in turn aggregates that `PduTriggering`.]()

[constr_3254] `FrameTriggering.framePort` reference condition [A `FrameTriggering` shall only reference a `FramePort` if the `CommunicationConnector` aggregating that `FramePort` is referenced by the `PhysicalChannel` which in turn aggregates that `FrameTriggering`.]()

[constr_3255] `FrameTriggering.pduTriggering` reference condition with regard to the `PhysicalChannel` [A `FrameTriggering` shall only reference a `PduTriggering` in the role `pduTriggering` if both the `FrameTriggering` and `PduTriggering` are aggregated by the same `PhysicalChannel`.]()

[constr_3256] `PduTriggering.iSignalTriggering` reference condition with regard to the `PhysicalChannel` [A `PduTriggering` shall only reference an `ISignalTriggering` in the role `iSignalTriggering` if both the `PduTriggering` and `ISignalTriggering` are aggregated by the same `PhysicalChannel`.]()

The following rules apply for the creation of `PduTriggerings` and `IPduPorts`:

- **[TPS_SYST_01052] Routing of `UserDefinedPdus`, `NmPdus`, `NPdus`** [`UserDefinedPdus`, `NmPdus`, `NPdus` which are not going through the `PduRouter` get their triggering information via the containing `FrameTriggering` and `FramePort` (no `PduTriggering` is defined for these `Pdus`).]()

- [TPS_SYST_03021] Routing of **GeneralPurposePdus** with category **GLOBAL_TIME** [GeneralPurposePdus with category GLOBAL_TIME shall have PduTriggering and IPduPorts defined.]()
- [TPS_SYST_02091] Routing of **GeneralPurposePdus** with category **SD** and **GeneralPurposePdus** with category **DoIP** [GeneralPurposePdus with category SD and GeneralPurposePdus with category DoIP shall have PduTriggering and IPduPorts defined since no Frames and FrameTriggerings are defined for Pdus that are handled by the SoAd.]()
- [TPS_SYST_01053] Low-level routing of **NPdus** [In case of a low-level routing of NPdus the Pdus are handled like IPdus and the PduTriggering and IPduPort shall be defined.]()
- [TPS_SYST_01138] Low-level routing of **XcpPdus** [Low-level routing of GeneralPurposeIPdus with category XCP: In case of a low-level routing of GeneralPurposeIPdus with category XCP the Pdus are handled like IPdus and the PduTriggering and IPduPort shall be defined.]()
- [TPS_SYST_01054] Routing of **DcmIPdus** [DcmIPdus shall have PduTriggering and IPduPorts since they are handled by the PduR (connection to the Dcm and/or DcmIPdu-routing).]()
- [TPS_SYST_01055] Routing of **ISignalIPdus** that are part of a **MultiplexedIPdu** [ISignalIPdus that are part of a MultiplexedIPdu (static or dynamic) and are also handled by the Com module shall have a PduTriggering and IPduPorts since they are handled by the PduR (and Com). Especially it is allowed to ignore certain received parts of a MultiplexedIPdu in a specific ECU.]()
- [TPS_SYST_01056] Routing of **ISignalIPdus**, **UserDefinedIPdus**, **MultiplexedIPdus**, **GeneralPurposeIPdus**, **ContainerIPdus** [ISignalIPdus (not part of MultiplexedIPdus), UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus and ContainerIPdus shall have a PduTriggering and IPduPort if they are handled by the PduR. Especially it is allowed to ignore a certain IPdu out of a Flexray frame if it is not considered in a specific ECU.](RS_SYST_00055)
- [TPS_SYST_01057] Routing of **NmPdus** [If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com the NmPdu shall also be referenced by a corresponding PduTriggering where attribute iPduPort exists accordingly.]()
- [TPS_SYST_02059] Routing of **SecuredIPdus** [SecuredIPdus shall have a PduTriggering and IPduPort defined since they are handled by the PduR. Pdus that are part of a SecuredIPdu and are also handled by the Com module shall have a PduTriggering and IPduPorts since they are handled by the PduR (and Com).](RS_SYST_00054)

- [TPS_SYST_02061] Routing of IPdus that are part of a ContainerIPdu [IPdus that are part of a ContainerIPdu shall have a PduTriggering and IPduPorts since they are handled by the PduR.](RS_SYST_00055)

The following rule applies to the creation of ISignalTriggering and ISignalPort:

[TPS_SYST_01058] Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content [In case of a Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content, the reference between the ISignalTriggerings (that are referred to by the PduTriggering in the role iSignalTriggering) and the respective ISignalPorts shall not be created.]()

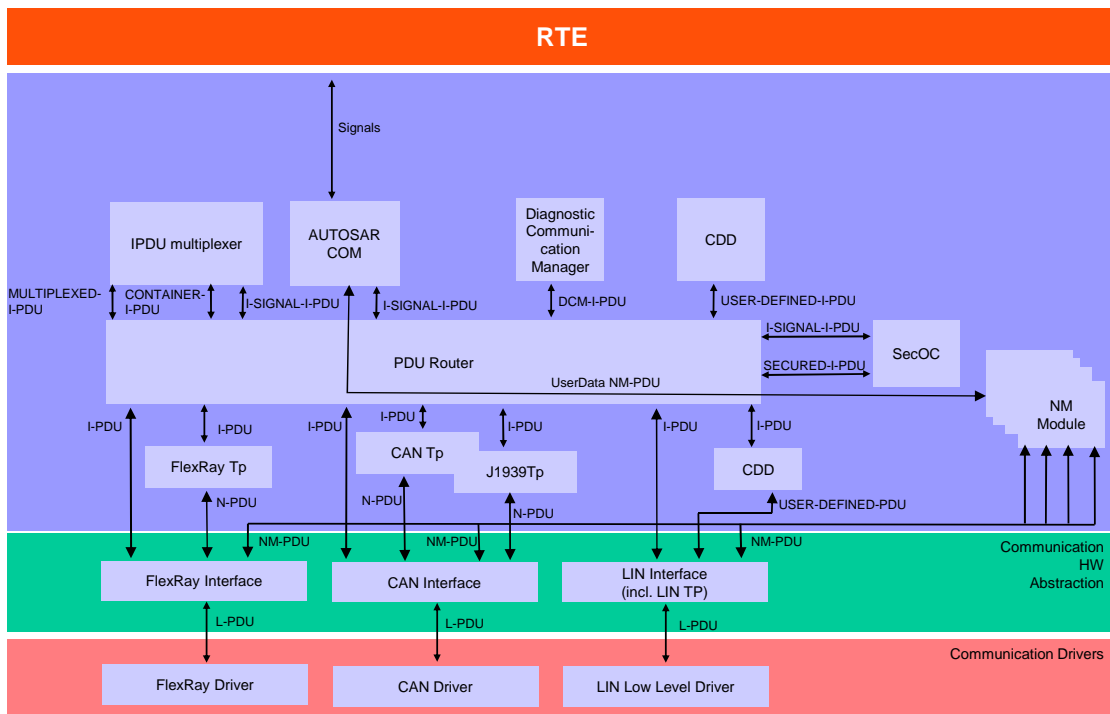


Figure 6.3: AUTOSAR Layered Architecture

Class	<i>CommConnectorPort</i> (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	The Ecu communication relationship defines which signals, Pdus and frames are actually received and transmitted by this ECU. For each signal, Pdu or Frame that is transmitted or received and used by the Ecu an association between an ISignalPort, IPduPort or FramePort with the corresponding Triggering shall be created. An ISignalPort shall be created only if the corresponding signal is handled by COM (RTE or Signal Gateway). If a Pdu Gateway ECU only routes the Pdu without being interested in the content only a FramePort and an IPduPort needs to be created.
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>





Class	CommConnectorPort (abstract)			
Subclasses	FramePort, IPduPort, ISignalPort			
Attribute	Type	Mult.	Kind	Note
communication Direction	Communication DirectionType	1	attr	Communication Direction of the Connector Port (input or output Port).

Table 6.1: CommConnectorPort

Class	FramePort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Connectors reception or send port on the referenced channel referenced by a FrameTriggering.			
Base	ARObject, CommConnectorPort, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.2: FramePort

Class	IPduPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Connectors reception or send port on the referenced channel referenced by a PduTriggering.			
Base	ARObject, CommConnectorPort, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
iPduSignal Processing	IPduSignalProcessing Enum	0..1	attr	Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.
rxSecurity Verification	Boolean	0..1	attr	This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.
timestampRx Acceptance Window	TimeValue	0..1	attr	This attribute is used to define the maximum allowed deviation in seconds from the expected timestamp for which a SecuredIPdu is still deemed authentic. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
useAuthData Freshness	Boolean	0..1	attr	This attribute describes whether a part of AuthenticPdu contained in a SecuredIPdu shall be passed on to the SWC that verifies and generates the Freshness. The part of the Authentic-PDU is defined by the authData FreshnessStartPosition and authDataFreshnessLength.

Table 6.3: IPduPort

[constr_3137] IPduPort.rxSecurityVerification is configurable on the receiver side [The IPduPort.rxSecurityVerification attribute shall only be used in IPduPorts with the communicationDirection = in.]()

[constr_3138] IPduPort.rxSecurityVerification validness [The IPduPort.rxSecurityVerification information is only valid for SecuredIPdus.]()

[constr_3337] IPduPort.useAuthDataFreshness is configurable on the receiver side [The `IPduPort.useAuthDataFreshness` attribute shall only be used in `IPduPorts` with the `communicationDirection = in`.]()

[constr_3338] IPduPort.useAuthDataFreshness validness [The `IPduPort.useAuthDataFreshness` information is only valid for `SecuredIPdus`.]()

Enumeration	IPduSignalProcessingEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Definition of signal processing modes.
Literal	Description
deferred	The signal indications / confirmations are deferred. Tags: atp.EnumerationLiteralIndex=0
immediate	The signal indications / confirmations are performed. Tags: atp.EnumerationLiteralIndex=1

Table 6.4: IPduSignalProcessingEnum

Class	ISignalPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Connectors reception or send port on the referenced channel referenced by an <code>ISignalTriggering</code> . If different timeouts or <code>DataFilters</code> for <code>ISignals</code> need to be specified several <code>ISignalPorts</code> may be created.			
Base	<code>ARObject</code> , <code>CommConnectorPort</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
dataFilter	<code>DataFilter</code>	0..1	aggr	Optional specification of a signal COM filter at the receiver side in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full <code>DataMapping</code> exist for the <code>SystemSignal</code> this information may be available from a configured <code>ReceiverComSpec</code> . In this case the <code>ReceiverComSpec</code> overrides this optional specification.
firstTimeout	<code>TimeValue</code>	0..1	attr	<ul style="list-style-type: none"> <code>ISignalPort</code> with <code>communicationDirection = in</code>: Optional first timeout value in seconds for the reception of the <code>ISignal</code>. <code>ISignalPort</code> with <code>communicationDirection = out</code>: Optional first timeout value in seconds for transmission deadline monitoring.
handleInvalid	<code>HandleInvalidEnum</code>	0..1	attr	This attribute defines how invalidation is applied to the <code>ISignals</code> received in the context of this <code>ISignalPort</code> .
timeout	<code>TimeValue</code>	0..1	attr	<ul style="list-style-type: none"> <code>ISignalPort</code> with <code>communicationDirection = in</code>: Optional timeout value in seconds for the reception of the <code>ISignal</code> in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full <code>DataMapping</code> exist for the <code>SystemSignal</code> this information may be available from a configured <code>ReceiverComSpec</code>, in this case the timeout value in <code>ReceiverComSpec</code> overrides this optional timeout specification.





Class	ISignalPort		
			<p style="text-align: center;">△</p> <ul style="list-style-type: none"> ISignalPort with communicationDirection = out: Optional timeout value in seconds for transmission deadline monitoring in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec, in this case the timeout value in SenderComSpec overrides this optional timeout specification.

Table 6.5: ISignalPort

Enumeration	HandleInvalidEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	Strategies of handling the reception of invalidValue.
Literal	Description
dontInvalidate	Invalidation is switched off. Tags: atp.EnumerationLiteralIndex=0
external Replacement	Replace a received invalidValue. The replacement value is sourced from the externalReplacement. Tags: atp.EnumerationLiteralIndex=1
keep	The application software is supposed to handle signal invalidation on RTE API level either by Data ReceiveErrorEvent or check of error code on read access. Tags: atp.EnumerationLiteralIndex=2
replace	Replace a received invalidValue. The replacement value is specified by the initValue. Tags: atp.EnumerationLiteralIndex=3

Table 6.6: HandleInvalidEnum

[TPS_SYST_01059] Relationship between [FrameTriggering](#) and [CommConnectorPort](#) [For the reference between [FrameTriggering](#) and [FramePort](#) two approaches are supported:

- One to One relationship between [FrameTriggering](#) and [FramePort](#) per [EcuInstance](#)
- One [FramePort](#) per [communicationDirection](#) per [EcuInstance](#) exists and is referenced by all applicable [FrameTriggerings](#) (n to 1).

]()

[TPS_SYST_01060] Relationship between [PduTriggering](#) and [CommConnectorPort](#) [For the reference between [PduTriggering](#) and [IPduPort](#) two approaches are supported:

- One to One relationship between [PduTriggering](#) and [IPduPort](#) per [EcuInstance](#)
- One [IPduPort](#) per [communicationDirection](#) per [EcuInstance](#) exists and is referenced by all applicable [PduTriggerings](#) (n to 1).

]()

[TPS_SYST_01061] Relationship between `ISignalTriggering` and `CommConnectorPort` [For the reference between `ISignalTriggering` and `ISignalPort` two approaches are supported:

- One to One relationship between `ISignalTriggering` and `ISignalPort` per `EcuInstance`
- One `ISignalPort` per `communicationDirection` per `timeout` per `EcuInstance` exists and is referenced by all applicable `PduTriggerings` (n to 1).

]()

[TPS_SYST_02208] `ISignalPort.handleInvalid` defines the reception invalidation behavior [The attribute `ISignalPort.handleInvalid` defines the behavior during signal reception if the respective `ISignal`'s `invalidValue` is received. The `ISignal` is assigned to this `ISignalPort` via the `ISignalTriggering`.]()

[TPS_SYST_02209] Not defined `ISignalPort.handleInvalid` behavior [If the attribute `ISignalPort.handleInvalid` is not defined then the value `dontInvalidate` shall be assumed.]()

[constr_5053] Existence of `ISignalPort.handleInvalid` [If the `ISignalPort` has a `networkRepresentationProps.invalidValue` defined then the `ISignalPort.communicationDirection` shall equal `in`.]()

[constr_5054] `externalReplacement` not applicable for `ISignalPort.handleInvalid` [In the context of `ISignalPort.handleInvalid` the value `externalReplacement` shall not be used.]()

The action `externalReplacement` can only be implemented by the RTE. Thus it is required to have a ComSpec definition.

[TPS_SYST_02210] Data invalidation in case the `dataTypePolicy` is set to `override` or `legacy` [If the `dataTypePolicy` of an `ISignal` is set to `override` or `legacy`, the `ISignalPort.handleInvalid` attribute defines the data invalidation.]()

()

6.2 ISignals

`SystemSignals` can be defined independently of `CommunicationClusters` and are representing the `VariableDataPrototypes`, `ArgumentDataPrototypes`, `Triggers` and `ModeDeclarationGroupPrototypes` in the communication description.

The RTE supports a "signal fan-out" where the same signal (System Signal) is sent in different `IPdus` to multiple receivers. The Pdu Router supports the "PDU fan-out" where the same `IPdu` is sent to multiple destinations.

To support the "signal fan-out" `ISignals` and `ISignalGroups` are introduced. An `ISignal(ISignalGroup)` represents the `SystemSignal(SystemSignalGroup)` of the Interaction Layer.

In case of "signal fan-out", several `ISignals` in different `IPdus` refer to the same `SystemSignal`. The "Signal fan-out" will be executed by the RTE. `ISignals` describe the Interface between the precompile configured RTE and the potentially postbuild configured Com Stack.

The `ISignalToIPduMapping` element describes the mapping of `ISignals` to `ISignalIPdus` and defines the position of an `ISignal` within an `ISignalIPdu`.

[constr_3009] Overlapping of `ISignals` is prohibited [`ISignals` mapped to an `ISignalIPdu` shall not overlap.]()

[constr_3010] `ISignalIPdu` length shall not be exceeded [The combined length of all `ISignals` and `updateIndicationBitPositions` that are mapped into an `ISignalIPdu` shall not exceed the defined `Pdu length`.]()

[constr_3011] Overlapping of `updateIndicationBits` of `ISignals` is prohibited [The `updateIndicationBitPosition` for an `ISignal` in an `ISignalIPdu` shall not overlap with other `updateIndicationBitPositions` or `ISignal` locations.]()

[TPS_SYST_01062] Network representation of an `ISignal` [With the aggregation of `SwDataDefProps` in the role `networkRepresentationProps` the actual representation of the `ISignal` on the network can be specified.]([RS_SYST_00047](#))

[TPS_SYST_01063] Context of network representation of an `ISignal` [The `dataTypePolicy` defines from which context the network representation specification shall be taken.]([RS_SYST_00001](#), [RS_SYST_00047](#))

For an alternative network representation it is important to define an alternative `SwDataDefProps` especially `SwBaseType` defining alternative encoding (e.g. from float in `PortInterface` to integer on bus).

[constr_3060] Usage of `networkRepresentationProps` and `physicalProps` [Usage of `networkRepresentationProps` and `physicalProps` shall follow the restrictions given in table 6.7.]()

Attributes of SwDataDefProps	SystemSignal.physicalProps	ISignal.networkProps
additionalNativeTypeQualifier	NA	NA
annotation	NA	NA
baseType	NA	D
baseType.category	NA	M
BaseTypeDirectDefinition.baseTypeEncoding	NA	D
BaseTypeDirectDefinition.byteOrder	NA	NA
BaseTypeDirectDefinition.baseTypeSize	NA	0..1
BaseTypeDirectDefinition.memAlignment	NA	NA
BaseTypeDirectDefinition.nativeDeclaration	NA	NA
compuMethod	D	I
dataConstr	D	M
displayFormat	D	M
implementationDataType	NA	NA
invalidValue	NA	D
stepSize	NA	NA
swAddrMethod	NA	NA
swAlignment	NA	NA
swBitRepresentation	NA	NA
swCalibrationAccess	NA	NA
swCalprmAxisSet	NA	NA
swComparisonVariable	NA	NA
swDataDependency	NA	NA
swHostVariable	NA	NA
swImplPolicy	NA	NA
swIntendedResolution	NA	NA
swInterpolationMethod	NA	NA
swIsVirtual	NA	NA
swPointerTargetProps	NA	NA
swRecordLayout	NA	NA
swRefreshTiming	NA	NA
swTextProps	NA	NA
swValueBlockSize	NA	NA
unit	D	M
valueAxisDataType	NA	NA

Table 6.7: Allowed SwDataDefProps Attributes for the ISignal and SystemSignal

The following settings apply in table 6.7:

D Define the attribute independent from settings to the left.

I Inherit the definition from the left for usage in the scope of this element. This means that the information is taken over in the respective context without further ARXML configuration. The attribute of the SwDataDefProps shall not exist on the right side.

NA Attribute is **not applicable** for usage in the scope of this element.

M Attribute is **meaningless** in the scope of this element. As it was allowed in previous versions, declaring it as Not Applicable (NA) would break compatibility. Tools shall ignore such an attribute without a warning.

In case that the System Description doesn't use a complete Software Component Description (VFB View) the `physicalProps` and `networkRepresentationProps` are used to configure the Data Semantics.

The `networkRepresentationProps` contains a reference to the `SwBaseType`. This reference can be used for the derivation of the `ComSignalType` in the AUTOSAR Com Configuration.

[TPS_SYST_02001] `networkRepresentationProps` are mandatory in case the `dataTypePolicy` is set to `override` or `legacy` [If the `dataTypePolicy` of an `ISignal` is set to `override` or `legacy`, the `networkRepresentationProps` for the respective `ISignal` have to be specified.]()

[TPS_SYST_02006] Usage of `networkRepresentationFromComSpec` [If the `networkRepresentationFromComSpec` is used either the `SwDataDefProps` in the role `networkRepresentation` aggregated by the `SenderComSpec` or `ReceiverComSpec` shall exist or the `ImplementationDataType` shall exist.]()

Please note that some `categorys` of `CompuMethod` cannot be successfully converted to A2L [19] because A2L does not provide an equivalent semantics that comes close to the respective AUTOSAR semantics.

A prominent example for such a case is a `CompuMethod` of category `SCALE_LINEAR_AND_TEXTTABLE` that actually has more than one linear interval and a `texttable` part.

[TPS_SYST_02079] Identification of `ImplementationDataType` for a given `ISignal` in an Ecu Extract [

1. From the `ISignal` go to the referenced `SystemSignal`
2. Find all `DataMappings` that refer to the `SystemSignal`
3. For all `VariableDataPrototypes` referenced by the applicable `DataMappings`
 - (a) If the `VariableDataPrototype` is typed by an `ApplicationDataType` and belongs to a `CompositionSwComponentType` then for all `DataTypeMappingSets` referenced by the `CompositionSwComponentType` find the `DataTypeMap` that refers to this `ApplicationDataType`. The `DataTypeMap` also refers to the wanted `ImplementationDataType`.
 - (b) If the `VariableDataPrototype` is typed by an `ApplicationDataType` and belongs to an `AtomicSwComponentType` then for all `DataTypeMappingSets` referenced by the `InternalBehavior` of the `AtomicSwComponentType` find the `DataTypeMap` that refers to this `ApplicationDataType`. The `DataTypeMap` also refers to the wanted `ImplementationDataType`.

- (c) If the `VariableDataPrototype` is typed by an `ImplementationDataType` then the `ImplementationDataType` is the wanted one.

]()

[TPS_SYST_02076] networkRepresentationProps in case the dataTypePolicy is set to transformingISignal [If the value of `ISignal.dataTypePolicy` is set to `transformingISignal` then `ISignal.networkRepresentationProps` shall be ignored.]()

[constr_3199] ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation [In a complete model every `ISignal` that has `dataTypePolicy` set to `transformingISignal` shall reference a `DataTransformation`.]()

[TPS_SYST_01065] Mapping onto the ComSignalType enumeration [The mapping of `baseTypeSize`, `baseTypeEncoding`, `ISignal.iSignalType` and `SystemSignal.dynamicLength` onto the `ComSignalType` enumeration is described in [Table 6.8.](#)] (*RS_SYST_00029*)

In other words [Table 6.8](#) focuses only on the derivation of the `ComSignalType`. This table shall not be taken as a source to derive requirements on the modeling of `SwBaseTypes` used on the level of the RTE.

<i>BaseTypeEncoding</i>	<i>BaseTypeSize</i>	<i>ISignal.iSignal-Type</i>	<i>SystemSignal.dynami-cLength</i>	<i>ComSignalType</i>
2C	8 Bits	primitive	not applicable	SINT8, ComBit-Size derived from <i>ISignal.length</i>
2C	not available	primitive	not applicable	SINT8 if <i>ISignal.length</i> ≤ 8. ComBitSize derived from <i>ISignal.length</i>
2C	16 Bits	primitive	not applicable	SINT16, ComBit-Size derived from <i>ISignal.length</i>
2C	not available	primitive	not applicable	SINT16 if <i>ISignal.length</i> > 8 and ≤ 16. ComBitSize derived from <i>ISignal.length</i>
2C	32 Bits	primitive	not applicable	SINT32, ComBit-Size derived from <i>ISignal.length</i>
2C	not available	primitive	not applicable	SINT32 if <i>ISignal.length</i> > 16 and ≤ 32. ComBitSize derived from <i>ISignal.length</i>
2C	64 Bits	primitive	not applicable	SINT64, ComBit-Size derived from <i>ISignal.length</i>
2C	not available	primitive	not applicable	SINT64 if <i>ISignal.length</i> > 32 and ≤ 64. ComBitSize derived from <i>ISignal.length</i>
NONE	8 Bits	primitive	not applicable	UINT8, ComBit-Size derived from <i>ISignal.length</i>



△

<i>BaseTypeEncoding</i>	<i>BaseTypeSize</i>	<i>ISignal.iSignal-Type</i>	<i>SystemSignal.dynamicalLength</i>	<i>ComSignalType</i>
NONE	not available	primitive	not applicable	UINT8 if <i>ISignal.length</i> ≤ 8. ComBitSize derived from <i>ISignal.length</i>
NONE	16 Bits	primitive	not applicable	UINT16, ComBitSize derived from <i>ISignal.length</i>
NONE	not available	primitive	not applicable	UINT16 if <i>ISignal.length</i> > 8 and ≤ 16. ComBitSize derived from <i>ISignal.length</i>
NONE	32 Bits	primitive	not applicable	UINT32, ComBitSize derived from <i>ISignal.length</i>
NONE	not available	primitive	not applicable	UINT32 if <i>ISignal.length</i> > 16 and ≤ 32. ComBitSize derived from <i>ISignal.length</i>
NONE	64 Bits	primitive	not applicable	UINT64, ComBitSize derived from <i>ISignal.length</i>
NONE	not available	primitive	not applicable	UINT64 if <i>ISignal.length</i> > 32 and ≤ 64. ComBitSize derived from <i>ISignal.length</i>
IEEE754	32 Bits	primitive	not applicable	FLOAT32, ComBitSize derived from <i>ISignal.length</i>

▽

△

<i>BaseTypeEncoding</i>	<i>BaseTypeSize</i>	<i>ISignal.iSignal-Type</i>	<i>SystemSignal.dynamicalLength</i>	<i>ComSignalType</i>
IEEE754	not available	primitive	not applicable	FLOAT32, <i>ISignal.length</i> = 32. ComBitSize derived from <i>ISignal.length</i>
IEEE754	64 Bits	primitive	not applicable	FLOAT64, ComBitSize derived from <i>ISignal.length</i>
IEEE754	not available	primitive	not applicable	FLOAT64, <i>ISignal.length</i> = 64. ComBitSize derived from <i>ISignal.length</i>
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	8 Bits	array	false	UINT8_N, ComSignalLength derived from <i>ISignal.length</i>
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	not available	array	false	UINT8_N, ComSignalLength derived from <i>ISignal.length</i>
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	8 Bits	array	true	UINT8_DYN, ComSignalLength derived from <i>ISignal.length</i>
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	not available	array	true	UINT8_DYN, ComSignalLength derived from <i>ISignal.length</i>
BOOLEAN	ignored	primitive	not applicable	BOOLEAN

Table 6.8: SwBaseType to ComSignalType Mapping

The setting "not applicable" for an Attribute in [Table 6.8](#) means that no value shall be set for this Attribute. The setting "ignored" for an Attribute in [Table 6.8](#) means that any

value is accepted for this Attribute, but the value will be ignored in creation of the ECU configuration value file.

[constr_3258] Restriction on ISignal.length in case iSignalType is set to array [If ISignal.iSignalType is set to array then ISignal.length shall be a multiple of 8.]()

[TPS_SYST_02111] VariableDataPrototype in case ISignal.iSignalType is set to array [If ISignal.iSignalType is set to array the corresponding VariableDataPrototype shall boil down to an Array according to [TPS_SYST_02083], [TPS_SYST_02084], [TPS_SYST_02085] and [TPS_SYST_02089].]()

The invalidValue is aggregated by the SwDataDefProps element. The SwDataDefProps and the SwBaseType classes are described in more detail in the Software Component Template [5].

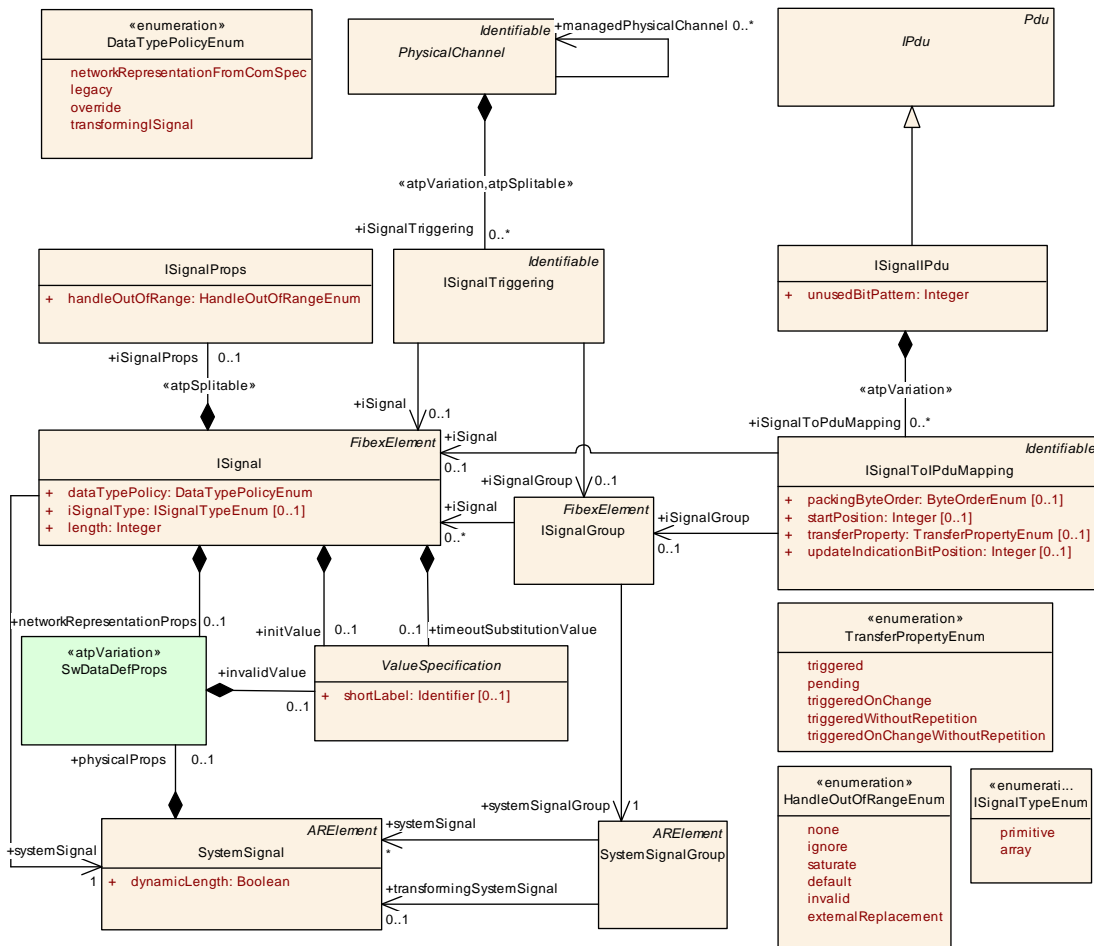


Figure 6.4: ISignals and the mapping into IPdus (FibexCore: SignalOverview)

The configuration of the COM Module for atomic signals can largely be derived from the System Template.

[TPS_SYST_01066] Derivation of Tx COM Signals [A `ComSignal` shall be defined in the COM module configuration for each `ISignalToIPduMapping` that is aggregated by `ISignalIPdu` that in turn is referenced by a `PduTriggering` that in turn references an `IPduPort` where the `communicationDirection` is set to `out` of the regarded ECU.

Exception: If the `ISignal` is part of a Signal Gateway relation (`ISignalMapping.targetSignal` pointing to an `ISignalTriggering` referencing this `ISignal`) the creation of a `ComSignal` is not mandated if

- the `ISignal` does not point to a `SystemSignal` that is referenced by a `DataMapping` (application does not send the gatewayed signal content) or
- the `ISignal` points to a `SystemSignal` that is referenced by a `DataMapping` where `communicationDirection` equals `in` and destination `ISignalTriggering.iSignalPort.communicationDirection` equals `out` (application sends the gatewayed signal content) or
- the `ISignalToIPduMapping.iSignal.dataTypePolicy` is set to `legacy`.

In these cases the configuration of `ComGwMapping` can be done by means of `ComGwSourceDescription` and `ComGwDestinationDescription`. However it is possible to create `ComSignals` for the `ComGwSignal` approach as well (i.e., even if the application does not require access to the respective `SystemSignal`).]()

[TPS_SYST_01067] Derivation of Rx COM Signals [A `ComSignal` shall be defined in the COM module configuration for each `ISignalToIPduMapping` that is aggregated by `ISignalIPdu` that in turn is referenced by a `PduTriggering` that in turn references an `IPduPort` where the `communicationDirection` is set to `in` in the regarded ECU.

Exception: If the `ISignal` is part of a Signal Gateway relation (`ISignalMapping.sourceSignal` pointing to an `ISignalTriggering` referencing this `ISignal`) the creation of a `ComSignal` is not mandated if

- the `ISignal` does not point to a `SystemSignal` that is referenced by a `DataMapping` (application is not interested in the gatewayed signal content) or
- the `ISignal` points to a `SystemSignal` that is referenced by a `DataMapping` where `communicationDirection` equals `out` and source `ISignalTriggering.iSignalPort.communicationDirection` equals `in` (application is not interested in the gatewayed signal content) or
- the `ISignalToIPduMapping.iSignal.dataTypePolicy` is set to `legacy`.

In these cases the configuration of `ComGwMapping` can be done by means of `ComGwSourceDescription` and `ComGwDestinationDescription`. However it is possible to create `ComSignals` for the `ComGwSignal` approach as well (i.e., even if the application does not require access to the respective `SystemSignal`).]()

To support the AUTOSAR concept of composite data types the AUTOSAR COM layer provides signal groups. Every record or array element of a composite data type requires a `SystemSignal` for the transmission. But the RTE has to guarantee the consistent transmission of data.

[TPS_SYST_01153] Atomic transport of `SystemSignalGroups` [A `SystemSignalGroup` shall be transmitted and received consistently; therefore it provides data consistency for composite data types.]()

A `SystemSignalGroup` refers to a set of `SystemSignals` that shall always be kept together in a common `IPdu`. An `ISignalGroup` represents a `SystemSignalGroup` of the Interaction Layer. In the case of "signal fan-out", several `ISignalGroups` refer to the same `SystemSignalGroup`.

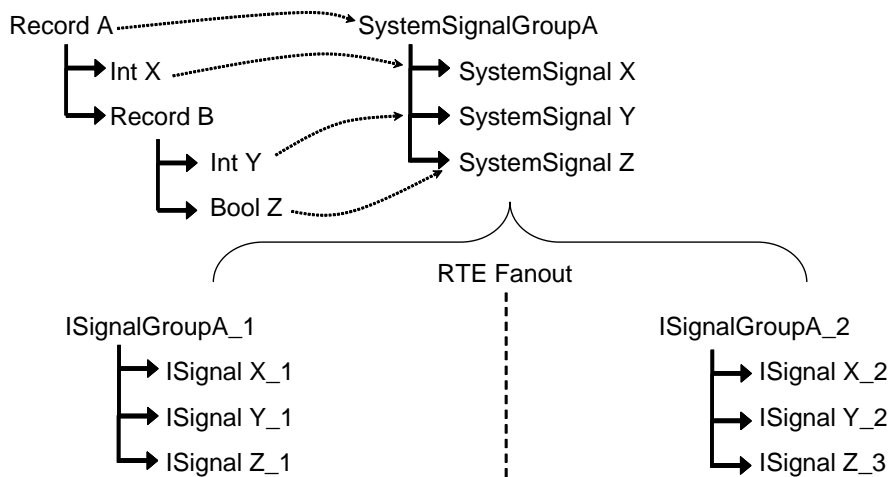


Figure 6.5: ISignal example

The example in Figure 6.5 shows the usage of `ISignalGroups` and `ISignals`. In this example a record is mapped to a `SystemSignalGroup`. All `ApplicationRecordElements` with `ApplicationPrimitiveDataType` are mapped to individual `SystemSignals`. If the same `SystemSignalGroup` is sent to different receivers (RTE Fanout) then two different `ISignalGroups` are created. For each `SystemSignal` within the `SystemSignalGroup` an `ISignal` is created. The different `ISignals` of the same `SystemSignal` can have different network representations.

[TPS_SYST_01156] Definition of `ISignalTriggerings` is allowed for `ISignalGroups` and for `GroupSignals` [If an `ISignalGroup` is referenced by an `ISignalTriggering` then the `ISignals` that are contained in the `ISignalGroup` (`GroupSignals`) may be referenced as well by `ISignalTriggerings`.]()

[constr_3094] Consistent `ISignalPort.communicationDirection` for `ISignalTriggerings` of `ISignalGroups` and contained `ISignals` [In case the `ISignals` contained in an `ISignalGroup` are referenced by an `ISignalTriggering`, the `communicationDirection` of the `ISignalPort` referenced by the

`ISignal`'s `ISignalTriggering` shall be identical to the `communicationDirection` of the `ISignalPort` referenced by the containing `ISignalGroup`'s `ISignalTriggering`.]()

Please note that not all `ISignals` that are part of the `ISignalGroup` need to have a reference to an `ISignalPort` via an `ISignalTriggering` as described by [TPS_SYST_02106].

[TPS_SYST_01157] **Allowed usage of attributes for `ISignals`, `ISignalGroups` and `GroupSignals`** [Table 6.9 shows attributes that may be used to configure `ISignals` in different roles (`ISignals` that are not part of an `ISignalGroup` and `ISignals` that are part of an `ISignalGroup`) and `ISignalGroups`.]()

Attributes	ISignal	ISignalGroup	GroupSignal
<code>startPosition</code>	1	NA	1
<code>updateIndicationBitPosition</code>	0..1	0..1	NA
<code>transferProperty</code>	0..1	0..1	0..1
<code>packingByteOrder</code>	1	NA	1
<code>dataFilter</code>	0..1	NA	0..1

Table 6.9: Allowed usage of attributes for `ISignals`, `ISignalGroups` and `GroupSignals`

[constr_3067] **`initValue` defined in the context of `ISignal`** [The definition of an `initValue` in the context of an `ISignal` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.]()

[constr_3437] **`invalidValue` defined in the context of `ISignal`** [The definition of `SwDataDefProps.invalidValue` aggregated by an `ISignal` in the role `networkRepresentationProps` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.]()

[constr_3438] **`timeoutSubstitutionValue` defined in the context of `ISignal`** [The definition of an `timeoutSubstitutionValue` in the context of an `ISignal` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.]()

[TPS_SYST_02012] **`initValue` and `invalidValue` represent internal values** [The `initValue` and `invalidValue` aggregated by the `networkRepresentationProps` shall represent the internal values.]()

[TPS_SYST_02110] **Default behavior for `ISignal.iSignalType`** [In case `ISignal.iSignalType` is not defined the value "primitive" shall be assumed.]()

[TPS_SYST_02144] ComTimeoutSubstitution does not apply for signal gateway operation [The specification of `ComTimeoutSubstitution` by defining the `ISignal.timeoutSubstitutionValue` does not apply for signal gateway operation. Only when the `ISignal` is processed for an upper layer the `ComTimeoutSubstitution` is actually performed.]()

Note: Since an `ISignal` may be candidate for both - local reception and gateway operation - a definition of `ISignal.timeoutSubstitutionValue` is valid on `ISignals` which are defined for gateway operation.

Class	ISignal			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different <code>SignalIPdus</code> to multiple receivers.</p> <p>To support the RTE "signal fan-out" each <code>SignalIPdu</code> contains <code>ISignals</code>. If the same System Signal is to be mapped into several <code>SignalIPdus</code> there is one <code>ISignal</code> needed for each <code>ISignalToIPduMapping</code>.</p> <p><code>ISignals</code> describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping).</p> <p>In case of the <code>SystemSignalGroup</code> an <code>ISignal</code> shall be created for each <code>SystemSignal</code> contained in the <code>SystemSignalGroup</code>.</p> <p>Tags:atp.recommendedPackage=ISignals</p>			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
data Transformation	DataTransformation	0..1	ref	<p>Optional reference to a <code>DataTransformation</code> which represents the transformer chain that is used to transform the data that shall be placed inside this <code>ISignal</code>.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataTransformation, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p>
dataTypePolicy	DataTypePolicyEnum	1	attr	<p>With the aggregation of <code>SwDataDefProps</code> an <code>ISignal</code> 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.</p> <p>If the policy "networkRepresentationFromComSpec" is chosen the network representation from the <code>ComSpec</code> that is aggregated by the <code>PortPrototype</code> shall be used. If the "override" policy is chosen the requirements specified in the <code>PortInterface</code> and in the <code>ComSpec</code> are not fulfilled by the <code>networkRepresentationProps</code>. In case the <code>SystemDescription</code> doesn't use a complete <code>SoftwareComponentDescription</code> (VFB View) the "legacy" policy can be chosen.</p>
initValue	ValueSpecification	0..1	aggr	<p>Optional definition of a <code>ISignal</code>'s <code>initValue</code> in case the <code>SystemDescription</code> doesn't use a complete <code>SoftwareComponentDescription</code> (VFB View). This supports the inclusion of legacy system signals.</p> <p>This value can be used to configure the <code>Signal</code>'s "Init Value".</p>





Class	ISignal			
				<p>△</p> <p>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.</p>
iSignalProps	ISignalProps	0..1	aggr	<p>Additional optional ISignal properties that may be stored in different files.</p> <p>Stereotypes: atpSplittable Tags:atp.Splitkey=iSignalProps</p>
iSignalType	ISignalTypeEnum	0..1	attr	<p>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.</p>
length	Integer	1	attr	<p>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.</p> <p>The ISignal length of zero bits is allowed.</p>
network Representation Props	SwDataDefProps	0..1	aggr	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
systemSignal	SystemSignal	1	ref	<p>Reference to the System Signal that is supposed to be transmitted in the ISignal.</p>
timeout Substitution Value	ValueSpecification	0..1	aggr	<p>Defines and enables the ComTimeoutSubstitution for this ISignal.</p>
transformation ISignalProps	TransformationISignalProps	*	aggr	<p>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.</p>

Table 6.10: ISignal

Enumeration	DataTypePolicyEnum
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping
Note	This class lists the supported DataTypesPolicies.
Literal	Description
legacy	In case the System Description doesn't use a complete Software Component Description (VFB View) this value can be chosen. This supports the inclusion of legacy signals. The aggregation of SwDataDefProps shall be used to configure the "ComSignalDataInvalidValue" and the Data Semantics. Tags: atp.EnumerationLiteralIndex=0
network Representation FromComSpec	Ignore any networkRepresentationProps of this ISignal and use the networkRepresentation from the ComSpec. Please note that the usage does not imply the existence of the SwDataDefProps in the role network Representation aggregated by the SenderComSpec or ReceiverComSpec if an ImplementationData Type is defined. Tags: atp.EnumerationLiteralIndex=1
override	If this value is chosen the requirements specified in the ComSpec (networkRepresentationFromComSpec) are not fulfilled by the aggregated SwDataDefProps. In this case the networkRepresentation is specified by the aggregated swDataDefProps. Tags: atp.EnumerationLiteralIndex=2
transformingISignal	This literal indicates that a transformer chain shall be used to communicate the ISignal as UINT8_N over the bus. Tags: atp.EnumerationLiteralIndex=4

Table 6.11: DataTypePolicyEnum

Enumeration	ISignalTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	This enumeration defines ISignal types that are used for derivation of the ComSignalType in the COM configuration.
Literal	Description
array	ISignal shall be interpreted as an array (UINT8_N, UINT8_DYN) Tags: atp.EnumerationLiteralIndex=0
primitive	ISignal shall be interpreted as a primitive type (e.g. UINT_8, SINT_32) Tags: atp.EnumerationLiteralIndex=1

Table 6.12: ISignalTypeEnum

Class	ISignalProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Additional ISignal properties that may be stored in different files.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
handleOutOfRange	HandleOutOfRangeEnum	1	attr	This attribute defines the outOfRangeHandling for received and sent signals.

Table 6.13: ISignalProps

Enumeration	HandleOutOfRangeEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	A value of this type is taken for controlling the range checking behavior of the AUTOSAR RTE.
Literal	Description
default	The RTE will use the initValue if the actual value is out of the specified bounds. Tags: atp.EnumerationLiteralIndex=0
external Replacement	This indicates that the value replacement is sourced from the attribute replaceWith. Tags: atp.EnumerationLiteralIndex=1
ignore	The RTE will ignore any attempt to send or receive the corresponding dataElement if the value is out of the specified range. Tags: atp.EnumerationLiteralIndex=2
invalid	The RTE will use the invalidValue if the value is out of the specified bounds. Tags: atp.EnumerationLiteralIndex=3
none	A range check is not required. Tags: atp.EnumerationLiteralIndex=4
saturate	The RTE will saturate the value of the dataElement such that it is limited to the applicable upper bound if it is greater than the upper bound. Consequently, it is limited to the applicable lower bound if the value is less than the lower bound. Tags: atp.EnumerationLiteralIndex=5

Table 6.14: HandleOutOfRangeEnum

Class	ISignalGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<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>Tags:atp.recommendedPackage=ISignalGroup</p>			
Base	<p><i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i></p>			
Attribute	Type	Mult.	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>Stereotypes: atpSplitable; atpVariation Tags: 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.





Class	ISignalGroup			
transformation ISignalProps	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 ISignal Groups are described in the TransformationTechnology class.

Table 6.15: ISignalGroup

Class	SystemSignalGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>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.</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>Tags:atp.recommendedPackage=SystemSignalGroups</p>			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
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 6.16: SystemSignalGroup

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			
Attribute	Type	Mult.	Kind	Note
iSignal	ISignal	0..1	ref	<p>Reference to a ISignal that is mapped into the ISignal IPdu.</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	ISignalGroup	0..1	ref	<p>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.</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>





Class	ISignalToIPduMapping			
packingByte Order	ByteOrderEnum	0..1	attr	<p>This parameter defines the order of the bytes of the signal and the packing into the SignalPdu. 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 SignalPdu (see the startPosition attribute description).</p> <p>For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.</p>
startPosition	Integer	0..1	attr	<p>This parameter is necessary to describe the bitposition of a signal within an SignalPdu. 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>Defines how the referenced ISignal contributes to the send triggering of the ISignalPdu.</p>
update IndicationBit Position	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 SignalPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored.</p> <p>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 ISignalPdu 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 6.17: ISignalToIPduMapping

[constr_3514] No two ISignalToIPduMappings shall reference the identical ISignal [No two ISignalToIPduMappings shall reference the identical ISignal in the role iSignal in the scope of one System.]()

Enumeration	TransferPropertyEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Transfer Properties of a Signal.
Literal	Description
pending	If the signal has the TransferProperty pending, then the function Com_SendSignal shall not perform a transmission of the IPdu associated with the signal. Tags: atp.EnumerationLiteralIndex=0
triggered	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. Tags: atp.EnumerationLiteralIndex=1
triggeredOnChange	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. Tags: atp.EnumerationLiteralIndex=2
triggeredOnChange WithoutRepetition	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. Tags: atp.EnumerationLiteralIndex=3
triggeredWithout Repetition	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. Tags: atp.EnumerationLiteralIndex=4

Table 6.18: TransferPropertyEnum

[TPS_SYST_02198] Applicable **transferProperty** for **ISignal** [If the **ISignalToIPduMapping** refers to an **ISignal** in the role **iSignal** then

- the **pending transferProperty** does not cause transmission of the **ISignalIPdu** if the **ISignal** is updated.
- if the **ISignalIPdu** has an **EventControlledTiming** aggregated at the **TransmissionModeTiming** then the **transferProperty** values
 - **triggered** and **triggeredWithoutRepetition** do cause immediate transmission of the **ISignalIPdu** if the **ISignal** is updated.
 - **triggeredOnChange** and **triggeredOnChangeWithoutRepetition** do cause immediate transmission of the **ISignalIPdu** if the **ISignal** is updated and has changed.

]()

[constr_3459] Applicable **transferProperty** for group signal [If the **ISignalToIPduMapping** refers to an **ISignal** in the role **iSignal** and this **ISignal** is referenced by an **ISignalGroup** in the role **iSignal** then the **ISignalToIPduMapping** of the **ISignal** shall either

- have **transferProperty pending** or **triggeredOnChange** defined, or
- have no **transferProperty** defined.

]()

[constr_3460] Full definition of `transferProperty` for group signal [If at least one of the `ISignals` belonging to an `ISignalGroup` has a `transferProperty` defined (via their respective `ISignalToIPduMapping`) then all other `ISignals` belonging to the same `ISignalGroup` shall have a `transferProperty` defined as well.]()

Note: **[constr_3460]** ensures that either

- no group signal has a `transferProperty` defined, then **[TPS_SYST_02199]** applies, or
- every group signal has a `transferProperty` defined, then **[TPS_SYST_02200]** applies.

[TPS_SYST_02199] Applicable `transferProperty` for `ISignalGroup` and no group signal has `transferProperty` defined [If the `ISignalToIPduMapping` refers to an `ISignalGroup` in the role `iSignalGroup` and the `ISignalIPdu` has an `EventControlledTiming` aggregated at the `TransmissionModeTiming` and none of the `ISignals` belonging to the `ISignalGroup` have a `transferProperty` defined (via their respective `ISignalToIPduMapping`) then if the `ISignalToIPduMapping` of the `ISignalGroup` has the `transferProperty`

- `pending` defined then an update of this `ISignalGroup` does not cause the transmission of the `ISignalIPdu`.
- `triggered` or `triggeredWithoutRepetition` defined then an update of this `ISignalGroup` does cause immediate transmission of the `ISignalIPdu`.
- `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` defined then an update of this `ISignalGroup` in combination with the change of any of the contained group signals does cause immediate transmission of the `ISignalIPdu`.

]()

[TPS_SYST_02200] Applicable `transferProperty` for `ISignalGroup` and group signals have `transferProperty` defined [If the `ISignalToIPduMapping` refers to an `ISignalGroup` in the role `iSignalGroup` and the `ISignalIPdu` has an `EventControlledTiming` aggregated at the `TransmissionModeTiming` and the `ISignals` belonging to the `ISignalGroup` have a `transferProperty` defined (via their respective `ISignalToIPduMapping`) then if the `ISignalToIPduMapping` of the `ISignalGroup` has the `transferProperty`

- `pending` defined then an update of this `ISignalGroup` does not cause the transmission of the `ISignalIPdu`.
- `triggered` or `triggeredWithoutRepetition` defined then an update of this `ISignalGroup` does cause immediate transmission of the `ISignalIPdu`.

- `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` defined then an update of this `ISignalGroup` in combination with the change of any of the contained group signals which have `transferProperty=triggeredOnChange` defined does cause immediate transmission of the `ISignalIPdu`.

]()

[constr_3461] TransferProperty for group signals if `ISignalGroup` has `transferProperty=pending` [If the `ISignalToIPduMapping` refers to an `ISignalGroup` in the role `iSignalGroup` and the `transferProperty` is set to `pending` then the group signals of this `ISignalGroup` shall either

- have no `transferProperty` defined (via their respective `ISignalToIPduMapping`) or
- every `ISignal` belonging to the `ISignalGroup` shall have the `transferProperty=pending` defined.

]()

Class	ISignalTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A <code>ISignalTriggering</code> allows an assignment of <code>ISignals</code> to physical channels.			
Base	<code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
<code>iSignal</code>	<code>ISignal</code>	0..1	ref	This reference shall be used if an <code>ISignal</code> is transported on the <code>PhysicalChannel</code> . This reference forms an XOR relationship with the <code>ISignalTriggering-ISignalGroup</code> reference.
<code>iSignalGroup</code>	<code>ISignalGroup</code>	0..1	ref	This reference shall be used if an <code>ISignalGroup</code> is transported on the <code>PhysicalChannel</code> . This reference forms an XOR relationship with the <code>ISignalTriggering-ISignal</code> reference.
<code>iSignalPort</code>	<code>ISignalPort</code>	*	ref	References to the <code>ISignalPort</code> on every ECU of the system which sends and/or receives the <code>ISignal</code> . References for both the sender and the receiver side shall be included when the system is completely defined.

Table 6.19: ISignalTriggering

6.2.1 Efficient COM for large data

AUTOSAR defines an alternative communication path between the RTE and the Communication Stack called Efficient COM for large data module (LdCom). The System Template does not define specific attributes which would distinguish whether the traditional Com or the LdCom shall be used. The idea behind this feature is rather that

- IF the LdCom module is integrated in an Ecu
- AND the specific interaction fulfills certain properties

- THEN LdCom shall be used.

Thus the usage of LdCom inside an ECU is project specific and is not derived from system description properties.

Note: even when all requirements for usage of LdCom are fulfilled it is not necessarily required to actually have an LdCom module inside the respective Ecu. It is rather a project specific decision whether LdCom module is integrated.

All of the following requirements need to be fulfilled in order to allow the usage of LdCom for the specific `ISignal` / `ISignalIPdu` combination.

[TPS_SYST_02015] LdCom: only one `ISignal` mapped to the `ISignalIPdu` [Only if exactly one `ISignal` is mapped into an `ISignalIPdu` and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02016] LdCom: only Transformer output and UINT8_N or UINT8_DYN supported [Only if

- the data type of the `ISignal` is either `UINT8_N` or `UINT8_DYN`
- or the `ISignal` has a reference to the `DataTransformation` in the role `dataTransformation`

and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02017] LdCom: Opaque `ISignalToIPduMapping.packingByteOrder` [Only if `packingByteOrder` has the value "Opaque" and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02018] LdCom: `ISignalToIPduMapping.startPosition` shall be 0 [Only if `ISignalToIPduMapping.startPosition` equals 0 (zero) and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02019] LdCom: `ISignalToIPduMapping.transferProperty` shall be triggered or triggeredWithoutRepetition for sent `ISignals` [Only if `ISignalToIPduMapping.transferProperty` equals `triggered` or `triggeredWithoutRepetition` for a Signal that is sent by the `EcuInstance` and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02020] LdCom: No `IPduTiming.minimumDelay` defined [The `ISignal` is mapped into an `ISignalIPdu`. Only if this `ISignalIPdu` does not contain an `IPduTiming` in `IPduTimingSpecification` which has `IPduTiming.minimumDelay` defined, the `ISignalIPdu` is sent by the `EcuInstance` and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02021] LdCom: `ISignalToIPduMapping.updateIndicationBitPosition` shall not be defined [Only if `ISignalToIPduMapping.updateIndicationBitPosition` is not defined and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02022] LdCom: Only the `transmissionModeTrueTiming` defined [The `ISignal` is mapped into an `ISignalIPdu`. Only if this `ISignalIPdu` has exactly the `TransmissionModeDeclaration.transmissionModeTrueTiming` defined (via `ISignalIPdu.iPduTimingSpecification`), the `ISignalIPdu` is sent by the `EcuInstance` and the LdCom module is present, this `ISignal` shall be handled by LdCom.]([RS_SYST_00049](#))

[TPS_SYST_02023] LdCom: `DataFilter` "always" if `TransmissionModeCondition` defined [The `ISignal` is mapped into an `ISignalIPdu`. If this `ISignalIPdu` has either

- no `TransmissionModeDeclaration.transmissionModeCondition` defined (via `ISignalIPdu.iPduTimingSpecification`) or
- `DataFilter.dataFilterType` is set to "always" for the `TransmissionModeCondition` of this `ISignalIPdu`.

and this `ISignalIPdu` is sent by the `EcuInstance` and the LdCom module is present, this `ISignal` shall be handled by LdCom.]([RS_SYST_00049](#))

[TPS_SYST_02024] LdCom: No `ModeDrivenTransmissionModeCondition` defined [The `ISignal` is mapped into an `ISignalIPdu`. Only if this `ISignalIPdu` has no `TransmissionModeDeclaration.modeDrivenTrueCondition` and `modeDrivenFalseCondition`, the `ISignalIPdu` is sent by the `EcuInstance` and the LdCom module is present, this `ISignal` shall be handled by LdCom.]([RS_SYST_00049](#))

[TPS_SYST_02025] LdCom: Only `EventControlledTiming` defined [The `ISignal` is mapped into an `ISignalIPdu`. Only if this `ISignalIPdu` has an `EventControlledTiming` (via `TransmissionModeTiming.eventControlledTiming`) and the LdCom module is present, this `ISignal` shall be handled by LdCom.]([RS_SYST_00049](#))

[TPS_SYST_02026] LdCom: Only `EventControlledTiming` with no repetition defined [The `ISignal` is mapped into an `ISignalIPdu`. Only if this `ISignalIPdu` has an `EventControlledTiming` (via `TransmissionModeTiming.eventControlledTiming`) with `EventControlledTiming.numberOfRepetitions = 0` defined and the LdCom module is present, this `ISignal` shall be handled by LdCom.]([RS_SYST_00049](#))

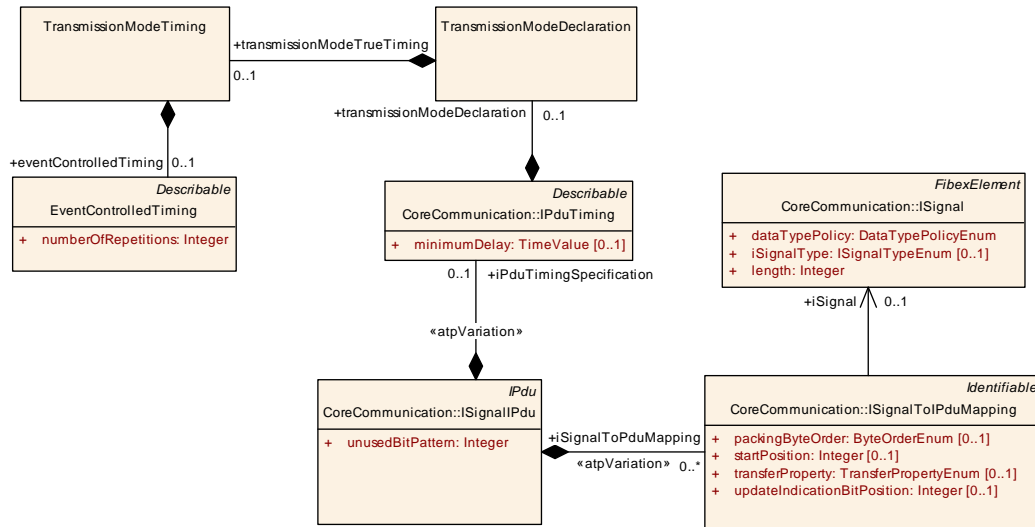


Figure 6.6: Pdu Timing excerpt that may be used to configure LdCom

[TPS_SYST_02027] LdCom: No `ISignalPort.timeout` reception timeout defined [Only if the `ISignalPort` which the `ISignalTriggering` is referring to has no `ISignalPort.timeout` defined and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02164] LdCom: No `ISignalPort.firstTimeout` reception timeout defined [Only if the `ISignalPort` which the `ISignalTriggering` is referring to has no `ISignalPort.firstTimeout` defined and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_02028] LdCom: No `ISignalPort.dataFilter` defined [Only if the `ISignalPort` which the `ISignalTriggering` is referring to has either

- no `ISignalPort.dataFilter` defined
- or the `DataFilter.dataFilterType` = always

and the LdCom module is present, this `ISignal` shall be handled by LdCom.] ([RS_SYST_00049](#))

[TPS_SYST_03001] LdCom: `ISignalIPdu` not part of any `ISignalIPduGroup` [Only if the `ISignalIPdu` is not referenced by any `ISignalIPduGroup` in the role `iSignalIPdu` and the LdCom module is present, this `ISignalIPdu` shall be handled by LdCom.] ([RS_SYST_00049](#))

The following table gives an overview about the specification item relevance for the sender and for the receiver side.

Spec Item Number	Spec Item Headline	Motivation	Applies to	Model element
[TPS_SYST_02015]	LdCom: only one <code>ISignal</code> mapped to the <code>ISignalIPdu</code> .	LdCom only supports one <code>ISignal</code> per PDU.	Rx + Tx	<code>ISignalToIPduMapping</code>
[TPS_SYST_02016]	LdCom: only Transformer output and <code>UINT8_N</code> or <code>UINT8_DYN</code> supported.	LdCom only supports Byte Array Signals.	Rx + Tx	<code>ISignal</code>
[TPS_SYST_02017]	LdCom: Opaque <code>ISignalToIPduMapping.packingByteOrder</code> .	LdCom only supports Byte Array Signals.	Rx + Tx	<code>ISignalToIPduMapping</code>
[TPS_SYST_02018]	LdCom: <code>ISignalToIPduMapping.startPosition</code> shall be 0.	LdCom only supports Signals starting in the first byte of the PDU.	Rx + Tx	<code>ISignalToIPduMapping</code>
[TPS_SYST_02019]	LdCom: <code>ISignalToIPduMapping.transferProperty</code> shall be triggered or triggeredWithoutRepetition.	LdCom only supports one <code>ISignal</code> per PDU. That signal has to trigger the transmission.	Tx	<code>ISignalToIPduMapping</code>
[TPS_SYST_02020]	LdCom: No <code>IPduTiming.minimumDelay</code> defined.	LdCom does not support event triggered transmission with repetitions.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02021]	LdCom: <code>ISignalToIPduMapping.updateIndicationBitPosition</code> shall not be defined.	LdCom does not support update bits.	Rx + Tx	<code>ISignalToIPduMapping</code>
[TPS_SYST_02022]	LdCom: Only the <code>transmissionModeTrueTiming</code> defined.	LdCom does not support transmission modes.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02023]	LdCom: <code>DataFilter</code> "always" if <code>TransmissionModeCondition</code> defined.	LdCom does not support transmission modes.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02024]	LdCom: No <code>ModeDrivenTransmissionModeCondition</code> defined.	LdCom does not support transmission modes.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02025]	LdCom: Only <code>EventControlledTiming</code> defined.	LdCom does not support cyclic transmission.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02026]	LdCom: Only <code>EventControlledTiming</code> with no repetition defined.	LdCom does not support cyclic transmission nor repeated transmission.	Tx	<code>ISignalIPdu</code>
[TPS_SYST_02027]	LdCom: No <code>ISignalPort.timeout</code> reception timeout defined.	LdCom does not support timeout handling.	Rx	<code>ISignalPort</code>
[TPS_SYST_02164]	LdCom: No <code>ISignalPort.firstTimeout</code> reception timeout defined.	LdCom does not support timeout handling.	Rx	<code>ISignalPort</code>
[TPS_SYST_02028]	LdCom: No <code>ISignalPort.dataFilter</code> defined.	LdCom does not support timeout handling.	Rx	<code>ISignalPort</code>
[TPS_SYST_03001]	LdCom: <code>ISignalIPdu</code> not part of any <code>ISignalIPduGroup</code> .	LdCom does not support <code>ISignalIPduGroups</code> .	Rx + Tx	<code>ISignalIPdu</code>

Table 6.20: Relevance of LdCom spec items for TX and RX

6.2.2 Big Endian and Little Endian memory layout of Pdus and Frames

The AUTOSAR system description provide means to specify how the memory layout looks like when signals are packed into `Pdus` and `Pdus` are packed into `Frames`. The layout of `Pdus` and `Frames` on different communication systems is out of scope of AUTOSAR. The specification of attributes Bit counting (monotone or sawtooth) and Bit order (decreasing or increasing)¹ is not supported by AUTOSAR. In AUTOSAR these attributes are fixed.

[TPS_SYST_01068] Bit Counting in AUTOSAR [The Bit counting shall always be considered as "sawtooth".]()

[TPS_SYST_01069] Bit Order in AUTOSAR [The bit order shall always be considered as "Decreasing".]()

¹More details about Bit counting and Bit order can be found in ASAM FIBEX [9].

When a signal is mapped into a `Pdu` only the `packingByteOrder` affects the memory layout of the signal inside the `Pdu` beginning with it's start bit position.

Little endian stores the least significant byte first and begins with the least significant bit, i.e. loworder bit in the sequence (the least significant bit serves as start bit).

Big endian stores the most significant byte first and begins with the most significant bit, i.e. the bit with the greatest numerical value (the most significant bit serves as start bit).

In both cases the bit positions in the mapped signals increase with the bit positions in the `ISignalIPdu` such that the bit 2^0 is mapped to position `n` in the `ISignalIPdu` and bit 2^1 is mapped to position `n+1` and so on.

Example 6.7 shows the memory layout for Little Endian and Big Endian if an `ISignal` with a length of 10 bits is mapped into a `Pdu`. The start bit position is 5.

Little Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	2^2	2^1	2^0	-	-	-	-	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3

Big Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	-	-	-	-

Figure 6.7: PackingByteOrder Example

The following examples are showing the mapping of Pdus into Frames.

The first example in Figure 6.8 for little endian shows a `Frame` with four bytes that contains a single `Pdu` that is two bytes long. The `PduToFrameMapping.startPosition` is defined with 8 and since the `packingByteOrder` is set to `mostSignificantByteLast` the `startPosition` denotes the least significant bit of the `Pdu` in the `Frame`. The bit position of the mapped `Pdu` increases with the bit positions in the `Frame` such that the bit 2^0 is mapped to position `n` in the `ISignalIPdu` and bit 2^1 is mapped to position `n+1` and so on.

Please note that the Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) for `PduToFrameMapping.startPosition` are allowed.

Figure 6.8 also shows that the `Pdu` contains three `ISignals`. The first `ISignal` has the `ISignalToIPduMapping.startPosition` defined as 0 and is 5 bits long. The bitposition of the second signal is 5 and the length is 10 bits. And the third signal has

the bitposition 15 and is only 1 bit long. Since the `ISignalToIPduMapping.packingByteOrder` is defined with `mostSignificantByteLast` as well the `startPosition` of the `ISignals` denotes the least significant bit of the `ISignal` in the `Pdu`.

Little Endian byte order:

Signal layout in Pdu:

Byte	1								2							
Bit	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16
Signal	2 ²	2 ¹	2 ⁰	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³
Pdu bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

Pdu layout in Frame:

Byte	0								1								2								3							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
Pdu	-	-	-	-	-	-	-	-	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	-	-	-	-	-	-	-	-

Figure 6.8: PackingByteOrder Example

The second example in [Figure 6.9](#) for big endian shows again a `Frame` with four bytes that contains a single `Pdu` that is two bytes long. The `PduToFrameMapping.startPosition` is defined with 15 and since the `packingByteOrder` is set to `mostSignificantByteFirst` the `startPosition` denotes the most significant bit of the `Pdu` in the `Frame`.

[Figure 6.9](#) also shows that the `Pdu` contains three `ISignals`. The first `ISignal` has the `ISignalToIPduMapping.startPosition` defined as 7 and is 5 bits long. The bitposition of the second signal is 2 and the length is 10 bits. And the third signal has the bitposition 8 and is only 1 bit long. Since the `ISignalToIPduMapping.packingByteOrder` is defined with `mostSignificantByteFirst` as well the `startPosition` of the `ISignals` denotes the most significant bit of the `ISignal` in the `Pdu`.

Big Endian byte order:

Signal layout in Pdu:

Byte	1								2							
Frame bit	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16
Signal	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁰
Pdu bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

Pdu layout in Frame:

Byte	0								1								2								3							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
Pdu	-	-	-	-	-	-	-	-	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	-	-	-	-	-	-	-	-

Figure 6.9: PackingByteOrder Example

Please note that the positioning of `SegmentPositions` in a `MultiplexedIPdu` works in the exact same way. The examples in [Figure 6.8](#) and [Figure 6.9](#) can be taken as well as an example for a `MultiplexedIPdu` where the 1 bit signal defines the `selectorField` and the other two signals represent segments defined for the `DynamicPart`.

6.3 PDUs

The chapter introduces the different [Pdu](#) types that are supported in the AUTOSAR Architecture and by the AUTOSAR Meta-Model.

The PDU Router is responsible only for the routing of [IPdus](#). Other [Pdus](#) that are direct specializations of the [Pdu](#) meta-class are not routed by the PDU Router.

[UserDefinedPdus](#) and [UserDefinedIPdus](#) are used to describe PDU-based communication over Complex Drivers. Chapter [6.13](#) provides a more detailed description of CDDs.

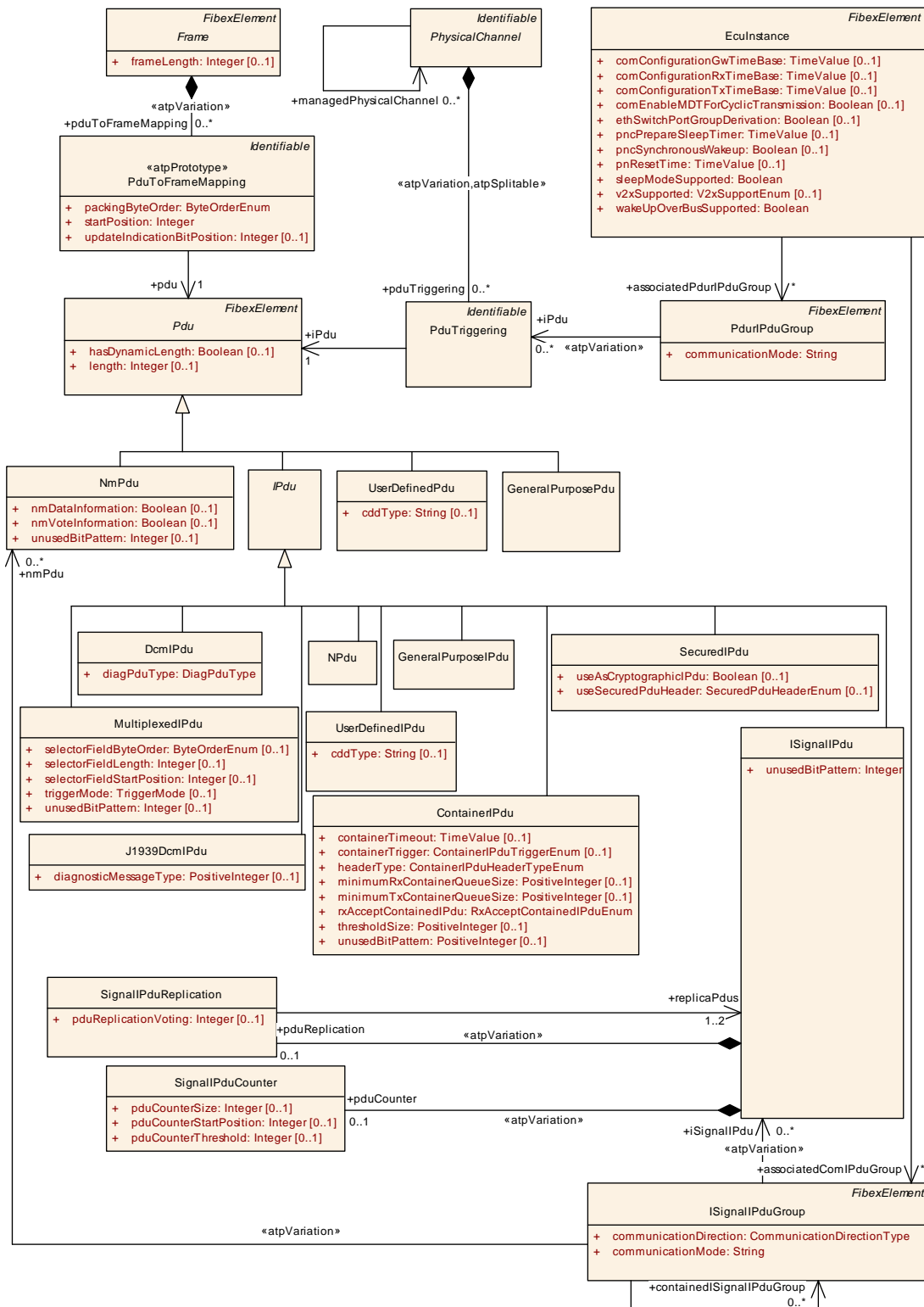


Figure 6.10: Pdu and the mapping into Frames (FibexCore: PDUOverview)

The `PduToFrameMapping` element describes the mapping of `Pdus` to `Frames` and defines the position of a `Pdu` within a `Frame`. By using different `PduToFrameMappings` it is possible to use the same `Pdu` in different `Frames`.

[constr_3516] limitation of `Frame.frameLength` for CAN L-PDUs [The `Frame.frameLength` of CAN PDUs shall be restricted to 0..8 for classic CAN L-PDUs and 0..8, 12, 16, 20, 24, 32, 48, 64 for CAN FD L-PDUs.]()

Please note that only a single `Pdu` is allowed to be mapped into a CAN Frame. The `Pdu.length` of the `Pdu` that is mapped into the CAN Frame is allowed to be smaller than the `Frame.frameLength` if padding is configured.

A timing description `IPduTiming` can be aggregated directly by the `ISignalIPdu`. This timing description can be used for the Configuration of COM Transmission Modes. The `PduTriggering` describes on which channel the `Pdu` is transmitted. Timing requirements may be specified with the Timing Extension model. More details are described in chapter 1.7.3. Such `Pdu` timing requirements needs to be fulfilled by the timing specification on the `Frame`.

Class	<code>Pdu</code> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Collection of all <code>Pdus</code> that can be routed through a bus interface.			
Base	<code>ARObject</code> , <code>CollectableElement</code> , <code>FibexElement</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PackageableElement</code> , <code>Referrable</code>			
Subclasses	<code>GeneralPurposePdu</code> , <code>IPdu</code> , <code>NmPdu</code> , <code>UserDefinedPdu</code>			
Attribute	Type	Mult.	Kind	Note
<code>hasDynamicLength</code>	Boolean	0..1	attr	This attribute defines whether the <code>Pdu</code> has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448] .
<code>length</code>	Integer	0..1	attr	<code>Pdu</code> length in bytes. In case of dynamic length <code>IPdus</code> (containing a dynamical length signal), this value indicates the maximum data length. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits. The <code>Pdu</code> length of zero bytes is allowed.

Table 6.21: Pdu

[constr_3448] Restriction for usage of `Pdu.hasDynamicLength` [The `Pdu.hasDynamicLength` attribute is only relevant for `UserDefinedPdus`, `UserDefinedIPdus`, `J1939DcmIPdus`.]()

For the remaining (I)`Pdus`, the fact whether they are dynamic or not should be derivable from the configuration of the upper layer:

- `ISignalIPdu`: At least one dynamic signal mapped to the `PDU`.
- `SecuredIPdu` and `MultiplexedIPdu`: At least one of the associated upper layer `PDU`s has dynamic length.
- `ContainerIPdu`: `Pdu` is dynamic if:
 - the `headerType` is `shortHeader` or `longHeader`.
 - the `headerType` is `noHeader` and the last contained `PDU` has dynamic length.

- **NPdu**: TP layer takes care of length handling, not visible to application.
- **DcmIPdu**: always dynamic
- **NmPdu**: always static
- **GeneralPurposePdu** and **GeneralPurposeIPdu**: Depending on upper layer, which could be: SD, TSync, DoIP, XCP, SomelpTp, Dlt.

Class	<i>IPdu</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Pdu</i> , <i>Referrable</i>			
Subclasses	<i>ContainerIPdu</i> , <i>DcmIPdu</i> , <i>GeneralPurposeIPdu</i> , <i>ISignalIPdu</i> , <i>J1939DcmIPdu</i> , <i>MultiplexedIPdu</i> , <i>NPdu</i> , <i>SecuredIPdu</i> , <i>UserDefinedIPdu</i>			
Attribute	Type	Mult.	Kind	Note
containedIPdu Props	ContainedIPduProps	0..1	aggr	Defines whether this IPdu may be collected inside a ContainerIPdu.

Table 6.22: IPdu

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	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>IPdu</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Pdu</i> , <i>Referrable</i>			
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: atpVariation Tags: 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: atpVariation Tags: vh.latestBindingTime=postBuild
pduCounter	SignalIPduCounter	0..1	aggr	An included Pdu counter is used to ensure that a sequence of Pdus is maintained. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime





Class	ISignalIPdu			
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>Stereotypes: atpVariation Tags: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 6.23: ISignalIPdu

Class	SignalIPduCounter			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A PduCounter is included in a predefined set of Pdus and used to ensure that a sequence of Pdus is maintained. The counter is incremented when a Pdu is transmitted. The receivers check if the received Pdu is the next one in sequence.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
pduCounterSize	Integer	0..1	attr	Size of PduCounter expressed in bits. Range: 1..8
pduCounterStartPosition	Integer	0..1	attr	Position of PduCounter expressed in bits. Note that Pdu Counter is not allowed to cross a byte border.
pduCounterThreshold	Integer	0..1	attr	Threshold value of IPduCounter algorithm. See AUTOSAR COM Spec for more details.

Table 6.24: SignalIPduCounter

Class	SignalIPduReplication			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	PduReplication 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 Pdu Counters, identical data content and are transmitted with the same frequency.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
pduReplicationVoting	Integer	0..1	attr	Number of identical IPdus needed for successful voting (1-3).
replicaPdus	ISignalIPdu	1..2	ref	Reference to replica Pdus of this IPdu.

Table 6.25: SignalIPduReplication

Class	NmPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>Network Management Pdu</p> <p>Tags:atp.recommendedPackage=Pdus</p>			





Class		NmPdu		
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
iSignalToIPdu Mapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUserData that is transmitted in the NmPdu. The counting of the startPosition starts at the beginning of the NmPdu regardless whether Cbv or Nid are used.
nmData Information	Boolean	0..1	attr	Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.
nmVote Information	Boolean	0..1	attr	Defines if the Pdu contains NM Vote information.
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM is filling not used areas of an Pdu with this bit-pattern. This attribute can only be used if the nmDataInformation attribute is set to true.

Table 6.26: NmPdu

Please note that in AUTOSAR only FrNm is able to send out NmPdus with and without voting information:

[constr_3073] nmVoteInformation only valid for FrNm [The [nmVoteInformation](#) attribute is only valid for FrNm.]()

Class		NPdu		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This is a Pdu of the Transport Layer. The main purpose of the TP Layer is to segment and reassemble IPdus. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.27: NPdu

Class		DcmIPdu		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by Dcm. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
diagPduType	DiagPduType	1	attr	Attribute is used to distinguish a request from a response.

Table 6.28: DcmIPdu

Enumeration	DiagPduType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Used to distinguish a diagnostic request from a response.
Literal	Description
diagRequest	Diagnostic Request Tags: atp.EnumerationLiteralIndex=0
diagResponse	Diagnostic Response Tags: atp.EnumerationLiteralIndex=1

Table 6.29: DiagPduType

Class	J1939DcmIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by J1939Dcm. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
diagnosticMessage Type	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table 6.30: J1939DcmIPdu

[constr_3096] Allowed values for diagnosticMessageType [The allowed values of diagnosticMessageType range from 1..57.]()

Class	GeneralPurposePdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is used for AUTOSAR Pdus without additional attributes that are routed by a bus interface. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.31: GeneralPurposePdu

[constr_3081] Value of category in **GeneralPurposePdu** [The attribute `category` of `GeneralPurposePdu` can have the following values:

- SD (Service Discovery)
- GLOBAL_TIME
- DoIP

]()

Class	GeneralPurposeIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.32: GeneralPurposeIPdu

[constr_3082] Value of category in **GeneralPurposeIPdu** [The attribute `category` of `GeneralPurposeIPdu` can have the following values:

- XCP
- SOMEIP_SEGMENTED_IPDU
- DLT

]()

Class	UserDefinedPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	UserDefinedPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the BusIf (e.g. a new Nm module) then this Pdu element shall be used to describe the communication. Tags: atp.recommendedPackage=Pdus			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedIPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table 6.33: UserDefinedPdu

Class	UserDefinedIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	UserDefinedIPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the PduR (e.g. a Diagnostic Service) then this IPdu element shall be used to describe the communication. Tags: atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable			
Attribute	Type	Mult.	Kind	Note
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table 6.34: UserDefinedIPdu

Class	<<atpPrototype>> PduToFrameMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A PduToFrameMapping defines the composition of Pdus in each frame.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
packingByteOrder	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the Pdu and the packing into the Frame. Please consider that [constr_3246] and [constr_3222] are restricting the usage of this attribute.
pdu	Pdu	1	ref	Reference to a I-Pdu, N-Pdu or NmPdu that is transmitted in the Frame.
startPosition	Integer	1	attr	This attribute describes the bitposition of a Pdu within a Frame. Please note that the absolute position of the Pdu in the Frame is determined by the definition of the packingByteOrder attribute. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the Frame. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the Frame. The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. The Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed.
updateIndicationBitPosition	Integer	0..1	attr	Indication to the receivers that the corresponding Pdu was updated by the sender. This attribute describes the position of the update bit in the frame that aggregates this PDUToFrameMapping. Length is always one bit. 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 Frame still undergoes a change.





Class	<<atpPrototype>> PduToFrameMapping			
				<p style="text-align: center;">△</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 6.35: PduToFrameMapping

[constr_3246] Frame.packingByteOrder mix within a Frame is not allowed [All PduToFrameMappings within a Frame shall have the same packingByteOrder value.]()

Please note that the absolute position (bit-position) of the Pdu in the Frame is determined by the definition of the packingByteOrder attribute. The Pdus are byte aligned in a Frame and the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed. For reasons of simplicity a mix of the packingByteOrder is not allowed.

[constr_3222] No ByteOrderEnum.opaque allowed for PduToFrameMapping.packingByteOrder [The values of PduToFrameMapping.packingByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.]()

Class	IPduTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES for each IPdu.</p> <p>The Transmission Mode of an IPdu that is valid at a specific point in time is selected using the values of the signals that are mapped to this IPdu. For each IPdu a Transmission Mode Selector is defined. The Transmission Mode Selector is calculated by evaluating the conditions for a subset of signals (class TransmissionModeCondition in the System Template).</p> <p>The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true and is defined to be false, if all Conditions evaluate to false.</p>			
Base	ARObject, Describable			
Attribute	Type	Mult.	Kind	Note
minimumDelay	TimeValue	0..1	attr	Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.
transmission Mode Declaration	TransmissionMode Declaration	0..1	aggr	AUTOSAR COM allows configuring statically two different transmission modes for each I-PDU (True and False). The Transmission Mode Selector evaluates the conditions for a subset of signals and decides the transmission mode. It is possible to switch between the transmission modes during runtime.

Table 6.36: IPduTiming

Class	PduTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu.</p> <p>Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface.</p> <p>If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.</p>			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
iPdu	Pdu	1	ref	<p>Reference to the Pdu for which the PduTriggering is defined. One I-Pdu can be triggered on different channels (PduR fan-out). The Pdu routing by the PduR is only allowed for subclasses of IPdu.</p> <p>Nevertheless is the reference to the Pdu element necessary since the PduTriggering element is also used to specify the sending and receiving connections to Ecu Ports.</p>
iPduPort	IPduPort	*	ref	<p>References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.</p> <p>References for both the sender and the receiver side shall be included when the system is completely defined.</p>
iSignal Triggering	ISignalTriggering	*	ref	<p>This reference provides the relationship to the ISignal Triggerings that are implemented by the PduTriggering. The reference is optional since no ISignalTriggering can be defined for DCM and Multiplexed Pdus.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild</p>





Class	PduTriggering			
secOcCrypto Mapping	SecOcCryptoService Mapping	0..1	ref	This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced Secured IPdu. Obviously, this reference is only applicable if the PduTriggering also references a SecuredIPdu in the role i Pdu.
triggerIPduSend Condition	TriggerIPduSend Condition	*	aggr	Defines the trigger for the Com_TriggerIPDUSend API call. Only if all defined TriggerIPduSendConditions evaluate to true (AND associated) the Com_Trigger IPDUSend API shall be called.

Table 6.37: PduTriggering

AUTOSAR COM provides a mechanism of starting/stopping COM PDU groups ([ISignalIPduGroup](#)). Please note that in a System Model an [ISignalIPdu](#) can belong to several [ISignalIPduGroups](#) of different [EcuInstances](#). So it is not possible to deduce the assignment of an [ISignalIPduGroup](#) to an [EcuInstance](#) via the [iSignalIPdus](#) that are included in the [ISignalIPduGroup](#). The assignment of an [ISignalIPduGroup](#) to an [EcuInstance](#) is therefore realized with the [EcuInstance.associatedComIPduGroup](#) reference.

Please note that the [EcuInstance.associatedComIPduGroup](#) reference is only used to assign the top-level [ISignalIPduGroups](#) to an [EcuInstance](#). The assignment of the [containedISignalIPduGroups](#) to an [EcuInstance](#) is done via the aggregating [ISignalIPduGroup](#) that in turn is referenced by an [EcuInstance](#).

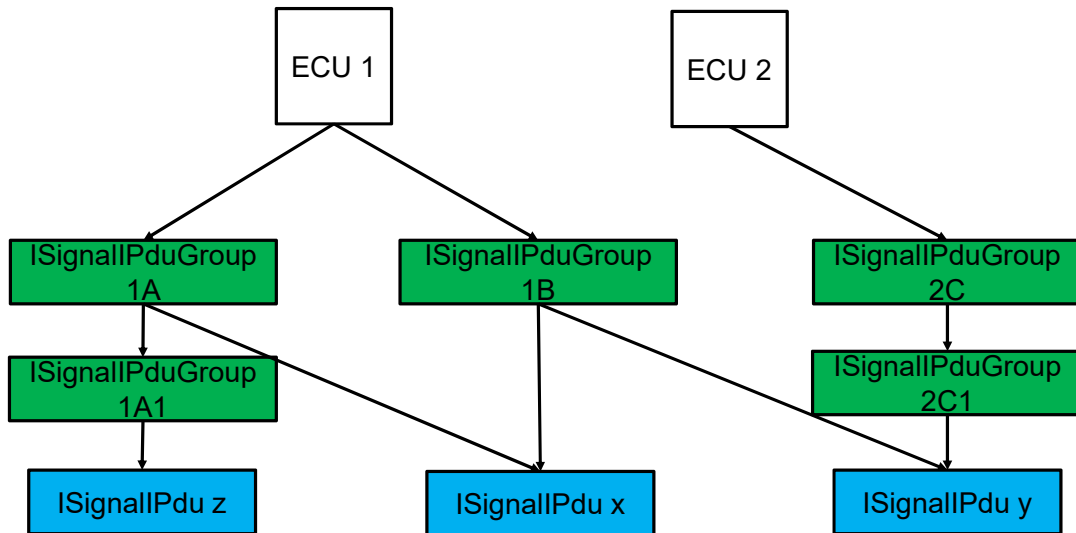


Figure 6.11: Example for ISignalIPduGroups and their assignment to EcuInstances

Class	ISignalPduGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalPduGroup contains either ISignalPdus or ISignalPduGroups. Tags: atp.recommendedPackage=ISignalPduGroup			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
communication Direction	Communication DirectionType	1	attr	This attribute determines in which direction IPdus that are contained in this IPduGroup will be transmitted (communication direction can be either In or Out).
communication Mode	String	1	attr	This attribute defines the use-case for this ISignalPdu Group (e.g. diagnostic, debugging etc.). For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
contained ISignalPdu Group	ISignalPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance.
iSignalPdu	ISignalPdu	*	ref	Reference to a set of Signal I-Pdus, which are contained in the ISignal I-Pdu Group. atpVariation: The content of a ISignal I-Pdu group can vary (->vehicle modes). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.38: ISignalPduGroup

Enumeration	CommunicationDirectionType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Describes the communication direction.
Literal	Description
in	Reception (Input) Tags: atp.EnumerationLiteralIndex=0
out	Transmission (Output) Tags: atp.EnumerationLiteralIndex=1

Table 6.39: CommunicationDirectionType
[constr_3020] communicationDirection of containedISignalIPduGroups

[The value of the attribute [communicationDirection](#) of [containedISignalIPduGroup](#) shall be identical to the value of the attribute [communicationDirection](#) of the enclosing [ISignalIPduGroup](#).]()

The AUTOSAR Pdu Router provides a mechanism of enabling/disabling of routing path groups ([PdurIPduGroup](#)).

Please note that in a System Model a `PduTriggering` can belong to several `PdurIPduGroups` of different `EcuInstances`. So it is not possible to deduce the assignment of an `PdurIPduGroup` to an `EcuInstance` via the `PduTriggerings` that are included in the `ISignalIPduGroup`. The assignment of an `PdurIPduGroup` to an `EcuInstance` is therefore realized with the `EcuInstance.associatedPdurIPduGroup` reference.

Class	PdurIPduGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The AUTOSAR PduR will enable and disable the sending of configurable groups of IPdus during runtime according to the AUTOSAR PduR specification. Tags: atp.recommendedPackage=PdurIPduGroups			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
communication Mode	<code>String</code>	1	attr	This attribute defines the use-case for this PdurIPdu Group. For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
iPdu	<code>PduTriggering</code>	*	ref	Reference to a set of IPdus, which are contained in the PduR I-Pdu Group. If an IPdu is routed by the PduR to different destinations (PduR fan-out) than an Pdu Triggering for each destination is created in the System Template. To enable/disable a specific destination the PdurIPduGroup refers to the PduTriggering. atpVariation: The content of a PduR I-Pdu group can vary (->vehicle modes). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.40: PdurIPduGroup

[constr_3275] `PduTriggering` containment in different `PdurIPduGroups` of the same `EcuInstance` is not allowed [A `PduTriggering` shall not be referenced by more than one `PdurIPduGroup` in the role `iPdu` where each of these `PdurIPduGroups` are referenced by the same `EcuInstance`.]()

6.3.1 ContainerIPdu

`IPdu` collection is used to transport several (smaller) `IPdus` in one (large) `ContainerIPdu`. A possible use case for example is the extended payload size for Ethernet and CanFd in combination with the limited payload of Can and Lin, where `Pdus` from a Can network shall be routed onto an Ethernet network and then back to a Can again.

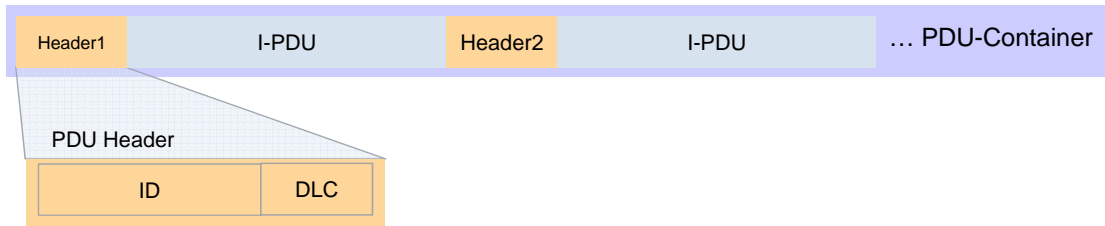


Figure 6.12: Layout of a ContainerIPdu if HeaderMode is used

For each `IPdu` which is put inside a `ContainerIPdu`, a header may be provided which determines which `IPdu` is contained (`ContainedIPduProps.headerIdLongHeader` or `headerIdShortHeader`) and what the size of that `IPdu` is (`DLC` during runtime). With this header mode the receivers are able to extract the individual contained `IPdus` again. As an alternative option to the usage of headers a statically configured layout of `IPdus` in the `ContainerIPdu` is supported.

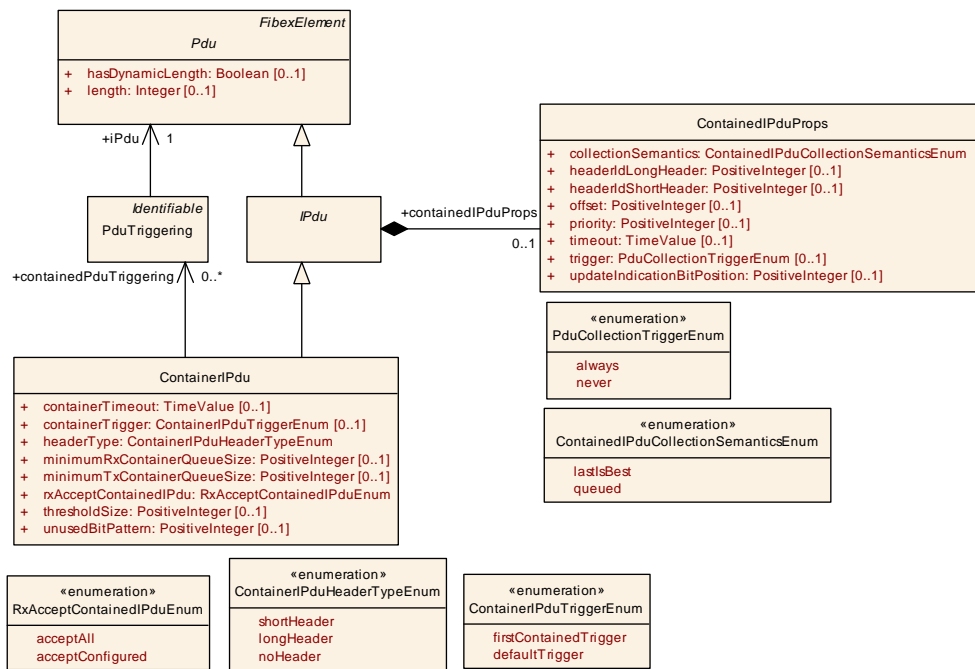


Figure 6.13: ContainerIPdu with ContainedIPduProps

Class	ContainerIPdu
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Allows to collect several IPdus in one ContainerIPdu based on the headerType. Tags: atp.recommendedPackage=Pdus
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable





Class	ContainerIPdu			
Attribute	Type	Mult.	Kind	Note
containedPduTriggering	PduTriggering	*	ref	This PduTriggering shall be collected inside the ContainerIPdu.
containerTimeout	TimeValue	0..1	attr	When this timeout expires the ContainerIPdu is sent out. The respective timer is started when the first Ipdu is put into the ContainerIPdu. This attribute is ignored on receiver side.
containerTrigger	ContainerIPduTriggerEnum	0..1	attr	Defines if the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into it. This attribute shall be ignored on receiver side.
headerType	ContainerIPduHeaderTypeEnum	1	attr	Defines whether and which header type is used (header id and length).
minimumRxContainerQueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for received containers.
minimumTxContainerQueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for transmitted containers.
rxAcceptContainedIPdu	RxAcceptContainedIPduEnum	1	attr	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
thresholdSize	PositiveInteger	0..1	attr	Defines the size threshold which, when exceeded, triggers the sending of the ContainerIPdu although the maximum Pdu size has not been reached yet. Unit: byte.
unusedBitPattern	PositiveInteger	0..1	attr	IPduM fills not updated areas of the ContainerPdu with this byte-pattern. Tags: atp.Status=draft

Table 6.41: ContainerIPdu

[constr_3436] Value range of [minimumTxContainerQueueSize](#) and [minimumRxContainerQueueSize](#) [If defined, the value of [minimumTxContainerQueueSize](#) and [minimumRxContainerQueueSize](#) shall be in the range of 0..255.]()

Enumeration	ContainerIPduTriggerEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines when the transmission of the ContainerIPdu shall be requested.
Literal	Description
defaultTrigger	Defines that the transmission of the ContainerIPdu shall be requested when the default trigger conditions apply (e.g. timeout of threshold). Tags: atp.EnumerationLiteralIndex=0
firstContainedTrigger	Defines that the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into the ContainerIPdu. Tags: atp.EnumerationLiteralIndex=1

Table 6.42: ContainerIPduTriggerEnum

Enumeration	ContainerIPduHeaderTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Is used to define the header type and size of ContainerIPdus. The header size includes the header id and the length information.
Literal	Description
longHeader	Header size is 64 bit: <ul style="list-style-type: none"> Header Id 32 bit Dlc 32 bit Tags: atp.EnumerationLiteralIndex=0
noHeader	No Header is used and the location of each containedPdu in the ContainerPdu is statically configured. Tags: atp.EnumerationLiteralIndex=2 atp.Status=draft
shortHeader	Header size is 32 bit: <ul style="list-style-type: none"> Header Id 24 bit Dlc 8 bit. Tags: atp.EnumerationLiteralIndex=1

Table 6.43: ContainerIPduHeaderTypeEnum

Enumeration	RxAcceptContainedIPduEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
Literal	Description
acceptAll	No fixed set of containedIPdus is defined for reception, any known containedIPdu (based on header Id) shall be expected within this ContainerIPdu. Tags: atp.EnumerationLiteralIndex=0
acceptConfigured	A fixed set of containedIPdus is defined for reception. Only these assigned containedIPdus (based on headerId) are expected in this ContainerIPdu. If a not assigned containedIPdu is received within this ContainerIPdu this containedIPdu is discarded. Tags: atp.EnumerationLiteralIndex=1

Table 6.44: RxAcceptContainedIPduEnum

Class	ContainedIPduProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Defines the aspects of an IPdu which can be collected inside a ContainerIPdu.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
collection Semantics	ContainedIPdu CollectionSemantics Enum	1	attr	Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.
headerIdLong Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.
headerIdShort Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.





Class	ContainedIPduProps			
offset	PositiveInteger	0..1	attr	Byte offset that describes the location of the Contained Pdu in the ContainerPdu if no header is used. Tags: atp.Status=draft
priority	PositiveInteger	0..1	attr	Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.
timeout	TimeValue	0..1	attr	Defines a IPdu specific sender timeout which can reduce the ContainerIPdu timer when this containedIPdu is put inside the ContainerIPdu. This attribute is ignored on receiver side.
trigger	PduCollectionTriggerEnum	0..1	attr	Defines whether this IPdu does trigger the sending of the ContainerIPdu. This attribute is ignored on receiver side.
update IndicationBit Position	PositiveInteger	0..1	attr	The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated. Tags: atp.Status=draft

Table 6.45: ContainedIPduProps

Enumeration	ContainedIPduCollectionSemanticsEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines the collection semantics for ContainedIPdus.
Literal	Description
lastIsBest	The ContainedIPdu data will be fetched via TriggerTransmit just before the transmission executes. Tags: atp.EnumerationLiteralIndex=0
queued	The ContainedIPdu data will instantly be stored to the ContainerIPdu in the context of the Transmit API. Tags: atp.EnumerationLiteralIndex=1

Table 6.46: ContainedIPduCollectionSemanticsEnum

Enumeration	PduCollectionTriggerEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	Defines whether a Pdu contributes to the triggering of the data transmission if Pdu collection is enabled.
Literal	Description
always	Pdu will trigger the transmission of the data. Tags: atp.EnumerationLiteralIndex=0
never	Pdu will be buffered and will not trigger the transmission of the data. Tags: atp.EnumerationLiteralIndex=1

Table 6.47: PduCollectionTriggerEnum

[TPS_SYST_02062] Allowed [ContainedIPduProps.headerIdLongHeader](#) and [ContainedIPduProps.headerIdShortHeader](#) values [[ContainedIPduProps.headerIdLongHeader](#) and [ContainedIPduProps.headerIdShortHeader](#) shall be restricted to values different from 0 (all bits of the value set to 0).] ([RS_SYST_00055](#))

Since the header information is larger than 8 bit the byte ordering of the header inside the [ContainerIPdu](#) needs to be defined. This is done at System level. Thus

all `ContainerIPdus` have the header information in the same byte order within one System.

[TPS_SYST_02063] Byte order of `ContainerIPdu` header information [The `System.containerIPduHeaderByteOrder` defines in which byte order the header information shall be put into the `ContainerIPdu`.] (*RS_SYST_00055*)

[constr_3140] No `ByteOrderEnum.opaque` allowed for `System.containerIPduHeaderByteOrder` [The values of `System.containerIPduHeaderByteOrder` are restricted to `ByteOrderEnum.mostSignificantByteFirst` and `ByteOrderEnum.mostSignificantByteLast`. I.e. the value `ByteOrderEnum.opaque` is not allowed.] ()

The following assumptions lead to the modeling of the `ContainerIPdu` structure:

- **[TPS_SYST_02097] Basic definition of contained `IPdus`** [Every `IPdu` with `IPdu.containedIPduProps` defined can be collected inside a `ContainerIPdu`.] (*RS_SYST_00055*)
- **[TPS_SYST_02098] Header id and header type of a contained `IPdu`** [A contained `IPdu` shall always have the same headerId per header type (long or short header), regardless in which `ContainerIPdu` it is collected. If `noHeader` is set then the contained `IPdu` does not need to have a headerId.] (*RS_SYST_00055*)
- **[TPS_SYST_02099] Relation between `ContainerIPdu` and contained `PduTriggerings` on sender side** [In the scope of one `EcuInstance` a `PduTriggering` shall be referenced by a `ContainerIPdu` in the role `containedPduTriggering` at most once.] (*RS_SYST_00055*)
- **[TPS_SYST_02100] Relation between `ContainerIPdu` and contained `IPdus` on receiver side** [On receiver side, it is not necessarily required to statically define which `IPdus` may be contained inside a `ContainerIPdu` if the header mode is used. Thus it would be possible to update the senders of `ContainerIPdus` and put different or additional `IPdus` inside.] (*RS_SYST_00055*)

The `ContainerIPdu` defines which `IPdus` may be collected inside that `ContainerIPdu` (`ContainerIPdu.containedPduTriggering`). Dynamic assignment of a contained `IPdu` to different `ContainerIPdus` during run-time is not supported by the `IPdu` multiplexer. Nevertheless it is allowed to collect an `IPdu` in several `ContainerIPdus` since each of those `ContainerIPdus` can be transmitted individually (on the same or on a different `PhysicalChannel`).

If a `ContainerIPdu` is transmitted on different `PhysicalChannels` this can be described in a System Description by a single `Pdu` element that is referenced by `PduTriggerings` of the different `PhysicalChannels`.

The content of the `ContainerIPdu` is described by `PduTriggerings` that are referenced in the role `containedPduTriggering`. Please note that although the `containedPduTriggerings` will also be transmitted on the same `PhysicalChannels` as the `ContainerIPdu` it is completely sufficient that the `containedPduTriggerings` that are referenced by the `ContainerIPdu` are aggregated by one of the

`PhysicalChannels` on which the `ContainerIPdu` will be transmitted. It is irrelevant which `PduTriggerings` of which `PhysicalChannels` are chosen to describe the content of the `ContainerIPdu` since the assignment of the `ContainerIPdu` to `PhysicalChannels` is done only by the `PduTriggerings` that are pointing to the `ContainerIPdu`.

The reason for this modeling is the configuration of the PduR. One use case in the PduR is the enabling and disabling of Pdus by `PdurIPduGroups`. The `PdurIPduGroup` works on `PduTriggerings` and there is also the use case to enable/disable the `ContainedIPdus` in the PduR.

[constr_3141] Only IPdus shall be part of a ContainerIPdu [The `PduTriggering` which is referenced in the role `ContainerIPdu.containedPduTriggering` shall refer to a subclass of an `IPdu` in the role `PduTriggering.ipdu`.]()

Only subclasses of `IPdus` are handled by the PduR and therefore are available for the `ContainerIPdu`.

For the sender side this assignment defines which `IPdus` may be collected inside this `ContainerIPdu`. For the receiver side this assignment may be omitted if `ContainerIPdu.rxAcceptContainedIPdu` is set to `RxAcceptContainedIPduEnum.acceptAll`.

[TPS_SYST_02064] Reception acceptance of contained IPdus [`ContainerIPdu.rxAcceptContainedIPdu` defines for the receiver side whether the list of `ContainerIPdu.containedPduTriggering` is a closed set.

If `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured`, only those `IPdus` (based on `headerId`) which are referenced by this `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` are extracted from this `ContainerIPdu`.

If `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll`, the `IPdus` (based on `headerId`) which are referenced by this `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` are expected inside this `ContainerIPdu` but also any other `IPdu` is extracted which is referenced by any other `ContainerIPdu.rxAcceptContainedIPdu` with `RxAcceptContainedIPduEnum.acceptAll` set.](*RS_SYST_00055*)

Thus all referenced `IPdus` which are referenced from `ContainerIPdu.rxAcceptContainedIPdu` with `RxAcceptContainedIPduEnum.acceptAll` form the set of `IPdus` which are considered for the reception of ANY `ContainerIPdu` with `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll`.

For the receiver side `ContainerIPdu.containedPduTriggering` may be omitted if `ContainerIPdu.rxAcceptContainedIPdu` is set to `RxAcceptContainedIPduEnum.acceptAll`. Such a `ContainerIPdu` will accept any of the contained `IPdus`

from the `acceptAll IPdu` set if they have the `headerId` (`headerIdShortHeader` or `headerIdLongHeader`) defined by the `ContainerIPdu.headerType` configured.

There are use-cases where several `IPdus` with identical content are transported on the vehicle networks. One motivation to design the communication structure like this is that the communication timing attributes may be different and thus need to be represented by different `ISignalIPdus` (i.e. different `ISignalIPdu.ipduTimingSpecification`, e.g. to support the transmission on Can and Fr networks with different transmission modes). When the content of the `IPdus` is identical they can be transported using the same header Id. Thus it is transparent for the receiver via which channel a specific `IPdu` arrived in case of `ContainerIPdu` with `acceptAll` setting.

In such a setup it may occur that several `PduTriggerings` for `IPdus` with identical header IDs are defined as candidates for the set of `IPdus` which are considered for the reception of `acceptAll ContainerIPdus`. In such a case there is additional information required to select which of these `PduTriggerings` shall be taken for the configuration and which shall be omitted.

The proposal is to use the `PduTriggering.ipduPort` relation to distinguish whether a specific `PduTriggering` shall be taken into account for the set of `IPdus` which are considered for the reception of `acceptAll ContainerIPdus`.

[constr_3453] Uniqueness of header Id in case of `acceptAll ContainerIPdu` [In case several `PduTriggerings` are referenced in the role `ContainerIPdu.containedPduTriggering` by `ContainerIPdus` with `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll` and the `PduTriggerings` refer to

- either the same `IPdu` (with `IPdu.containedIPduProps`)
- or different `IPdus` (with `IPdu.containedIPduProps`) and equal header Ids,

only one `PduTriggering` shall have a reference to an `IPduPort` with direction IN which is applicable for this `EcuInstance`.]()

Note: **[constr_3453]** regulates that in the set of `IPdus` which are considered for the reception of `acceptAll ContainerIPdus` each header Id only occurs once and thus allows a deterministic configuration of the `IPduM` module and runtime behavior.

[TPS_SYST_02196] `PduTriggering` is referenced by several `ContainerIPdus` [In case a `PduTriggering` is referenced in the role `ContainerIPdu.containedPduTriggering` by several `ContainerIPdus` with `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll` and has a reference to an `IPduPort` with direction IN then this `PduTriggering` shall only be considered once for the set of `IPdus` which are considered for the reception of `acceptAll ContainerIPdus`.](*RS_SYST_00055*)

[constr_3403] Usage of `ContainerIPdu.rxAcceptContainedIPdu` if `noHeader` is used [If the `ContainerIPdu.headerType` is set to `noHeader` then the `ContainerIPdu.rxAcceptContainedIPdu` attribute value shall be set to `acceptConfigured`.]()

[TPS_SYST_03014] Transmission triggering by the first contained IPdu put into a ContainerIPdu [The attribute `ContainerIPdu.containerTrigger` determines whether the transmission of a `ContainerIPdu` shall be requested when the first contained `IPdu` was put into the `ContainerIPdu`.

In case `containerTrigger` equals `firstContainedTrigger` the transmission of the `ContainerIPdu` shall be requested when the first contained `IPdu` is put into the `ContainerIPdu`.

In case `containerTrigger` equals `defaultTrigger` the transmission of the `ContainerIPdu` shall be requested when the other trigger conditions defined by the `ContainerIPdu` are fulfilled (e.g. `containerTimeout`, `thresholdSize`).] (*RS_SYST_00055*)

Note: This trigger condition is independent from `PduCollectionTriggerEnum.always` which is defined for specific `IPdus`. With the attribute `ContainerIPdu.containerTrigger = firstContainedTrigger` on the other hand, any contained `IPdu` will trigger the `ContainerIPdu` transmission.

Rationale for this trigger condition is the efficient usage (allow the `ContainerIPdus` to reach a certain fill level) of triggered transmission on time- (`containerTrigger` typically set to `firstContainedTrigger`) and event-driven (`containerTrigger` typically set to `defaultTrigger`) buses.

Each `IPdu` which shall be collected inside a `ContainerIPdu` shall provide the `IPdu.containedIPduProps`. This `ContainedIPduProps` defines a header `Id` per `ContainerIPdu.headerType` which shall be used in the Pdu header of the `ContainerIPdu` in case that the `headerType` is set to `shortHeader` or `longHeader`. In case that the `headerType` is set to `noHeader` the layout of `IPdus` in the `ContainerIPdu` is statically configured and no header `Id` is required.

[constr_3454] Unique headerIdLongHeader for acceptConfigured [For a `ContainerIPdu` with `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured` and `ContainerIPdu.headerType = longHeader` the following shall apply: All referenced `IPdus` (via `ContainerIPdu.containedPduTriggering`) shall have a unique `ContainedIPduProps.headerIdLongHeader` within the scope of this `ContainerIPdu`.] ()

[constr_3455] Unique headerIdShortHeader for acceptConfigured [For a `ContainerIPdu` with `ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured` and `ContainerIPdu.headerType = shortHeader` the following shall apply: All referenced `IPdus` (via `ContainerIPdu.containedPduTriggering`) shall have a unique `ContainedIPduProps.headerIdShortHeader` within the scope of this `ContainerIPdu`.] ()

Note: With **[constr_3454]** and **[constr_3455]** it is possible to have the same header `Id` value received in different `ContainerIPdus`. It just has to be guaranteed that in the scope of one `ContainerIPdu` the reception of header `Id` is unambiguous.

[constr_3142] Mandatory headerIdLongHeader for longHeader [For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = longHeader the IPdu.containedIPduProps.headerIdLongHeader shall be defined.]()

[constr_3143] Mandatory headerIdShortHeader for shortHeader [For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = shortHeader the IPdu.containedIPduProps.headerIdShortHeader shall be defined.]()

[constr_3402] Mandatory offset if noHeader is used [For each IPdu which is assigned to a ContainerIPdu in the role containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.]()

[constr_3404] Usage of ContainedIPduProps.updateIndicationBitPosition [ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.]()

[constr_3405] Dynamic Length IPdu inside of a static configured ContainerIPdu [Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with headerType set to noHeader) is allowed to be a dynamic length IPdu (i.e., a contained IPdu that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained IPdus of a ContainerIPdu with static container layout have to be static length IPdus.]()

[TPS_SYST_02065] Contained IPdu specific transmission timeout [The IPdu specific transmission timeout can be specified at ContainedIPduProps.timeout. If no ContainedIPduProps.timeout is provided the timeout from the ContainerIPdu shall be used (ContainerIPdu.containerTimeout).](RS_SYST_00055)

The case where neither the ContainerIPdu.containerTimeout nor the ContainedIPduProps.timeout is provided, will result in no time-based triggering of ContainerIPdus which might lead to long delays or no transmission at all if no other sending condition for this ContainerIPdu does occur (e.g. no further IPdu is collected inside this ContainerIPdu).

[TPS_SYST_02066] ContainerIPdu.thresholdSize [The attribute ContainerIPdu.thresholdSize defines the threshold when a ContainerIPdu shall be triggered for transmission. If the payload size of the ContainerIPdu exceeds the value of thresholdSize this ContainerIPdu shall be transmitted.](RS_SYST_00055)

Note: The ContainerIPdu.thresholdSize supports the definition of a transmission threshold which takes the data transmission model of the communication into account. Especially when operating with variable length IPdus, only the maximum length of these IPdus is defined in the System Description. Only having the maximumLength information it is not possible to derive a sensible threshold for the ContainerIPdu

this variable length `IPdu` is collected in. Thus a `ContainerIPdu` would wait for further contained `IPdus`. Using a transmission model it can be calculated that the average size contained `IPdu` will not fit into that `ContainerIPdu` anymore and provide this as a requirement in `ContainerIPdu.thresholdSize`.

Another use case for the `ContainedIPduProps` is to support the usage of optimized trigger transmit collection of `IPdus` in `ContainerIPdus`. Therefore it is necessary to distinguish between contained `IPdus` with `lastIsBest` (will be fetched via trigger transmit just before the transmission executes) and those with `queued` semantics (will instantly be stored in the context of the transmit API). This distinction is possible on the level of single contained `IPdus` with the attribute `collectionSemantics`.

For all intents and purposes, the different handling of contained `IPdus` depending on the semantics is supported by the attribute `ContainedIPduProps.collectionSemantics` that allows the individual setting of the intended semantics per contained `IPdu`.

[constr_3517] Consistent setting of `ContainedIPduProps.collectionSemantics` in the context of one `ContainerIPdu` [The value of the attribute `ContainedIPduProps.collectionSemantics` shall be identical for all contained `IPdus` within the context of a given `ContainerIPdu`.]()

[constr_3144] Mandatory `IPdu.containedIPduProps` for contained `IPdus` [For each `IPdu` which is assigned to a `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` the `IPdu.containedIPduProps` shall be defined.]()

`ContainedIPduProps` is optional and may be ignored in case the `IPdu` is not mapped into a `ContainerIPdu`. A use-case is that an `IPdu` is fan-out in the `PduR` and one `PduTriggering` is part of a `ContainerIPdu` while the other `PduTriggering` is directly transported via a bus interface.

[constr_3488] Value range of `ContainedIPduProps.priority` [If defined, the value of `ContainedIPduProps.priority` shall be in the range of 0..255.]()

[constr_3489] `ContainedIPduProps.priority` is only applicable if a `ContainerIPdu` header is used [`ContainedIPduProps.priority` is only applicable if the `headerType` of the `ContainerIPdu` is set to `shortHeader` or `longHeader`.]()

[constr_3490] `ContainedIPduProps.priority` is only applicable if `collectionSemantics` is set to `lastIsBest` [`ContainedIPduProps.priority` is only applicable if `ContainedIPduProps.collectionSemantics` is set to `lastIsBest`.]()

6.3.2 SecuredIPdu

AUTOSAR supports an authentication mechanism for critical data on the level of `Pdus`.

[TPS_SYST_02060] **SecuredIPdus** [**SecuredIPdu** shall be used to describe an **IPdu** that is protected against unauthorized manipulation and replay attacks.]([RS_SYST_00054](#))

Please note that several **SecuredIPdus** may exist with the same **dataId** value since the same data may be transported on different channels.

[TPS_SYST_02148] **Meaning of useAsCryptographicIPdu that is not set or set to false** [If **useAsCryptographicIPdu** is not set or set to false the **SecuredIPdu** contains the payload of an Authentic **IPdu** supplemented by additional Authentication Information (Freshness Counter and an Authenticator).]([RS_SYST_00054](#))

[TPS_SYST_02149] **Meaning of useAsCryptographicIPdu that is set to true** [If **useAsCryptographicIPdu** is set to true the **SecuredIPdu** contains the Authenticator for a payload that is transported in a separate message. The separate Authentic **IPdu** is described by the **PduTriggering** that is referenced with the **payload** reference.]([RS_SYST_00054](#))

The attribute **useAsCryptographicIPdu** decides whether one single **Pdu** or two **Pdus** are transferred on the communication bus. In either case always two **IPdus** shall be modeled:

- **SecuredIPdu** with a **PduTriggering**
- **payload IPdu** with a **PduTriggering**

[TPS_SYST_02172] **Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to false** [If the **useAsCryptographicIPdu** is set to false only the **SecuredIPdu** shall be

- mapped into a Frame by the **PduToFrameMapping** or
- assigned to **SocketConnectionBundle** or **SocketConnection** or
- assigned to **ContainerIPdu**

]([RS_SYST_00054](#))

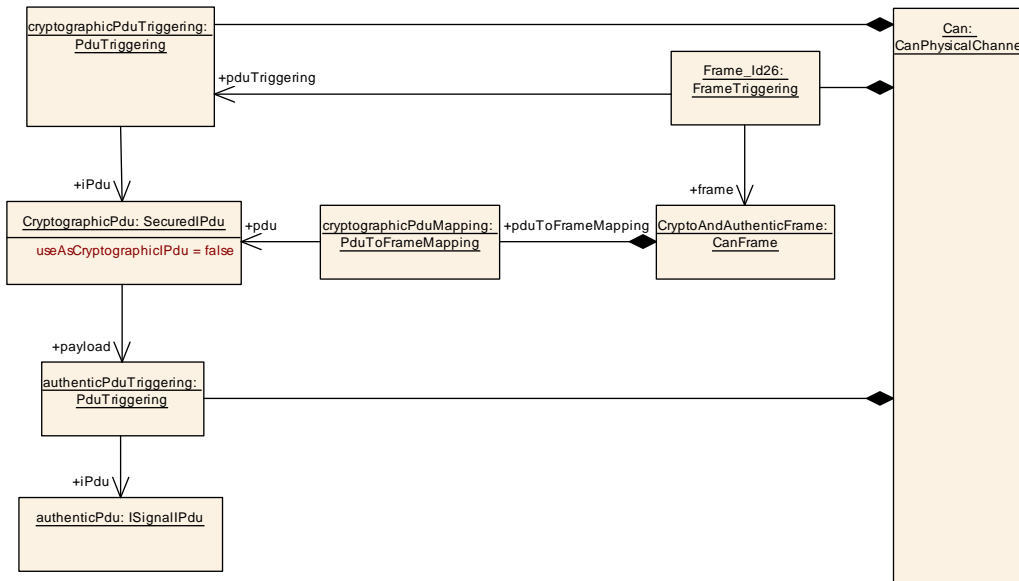


Figure 6.14: Example for the modeling of SecuredIPdu with useAsCryptographicIPdu set to false

[TPS_SYST_02173] Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to true [If the useAsCryptographicIPdu is set to true then the SecuredIPdu and the payload IPdu shall be

- mapped into a Frame by the PduToFrameMapping or
- assigned to SocketConnectionBundle or SocketConnection or
- assigned to ContainerIPdu

](RS_SYST_00054)

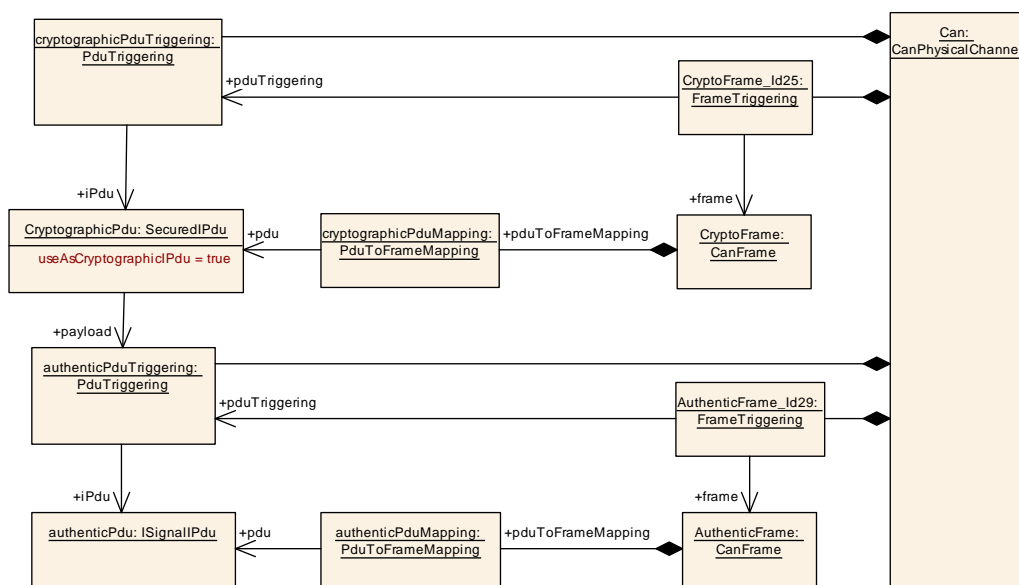


Figure 6.15: Example for the modeling of SecuredIPdu with useAsCryptographicIPdu set to true

Please note that [TPS_SYST_02059] defines that the `PduTriggerings` of the `SecuredIPdu` and `PduTriggerings` of the payload `IPdu` shall both reference `IPduPorts`.

A `SecuredIPdu` defines freshness properties by referencing the reusable `SecureCommunicationFreshnessProps` in the role `freshnessProps`. The authentication properties are defined by reusable `SecureCommunicationAuthenticationProps` that are referenced in the role `authenticationProps`. Configuration settings that are specific to the `SecuredIPdu` are defined in `SecureCommunicationProps`.

Please note that if the `SecuredIPdu` does not reference the `SecureCommunicationFreshnessProps` in the role `freshnessProps` the freshness value will not be included in the `SecuredIPdu`.

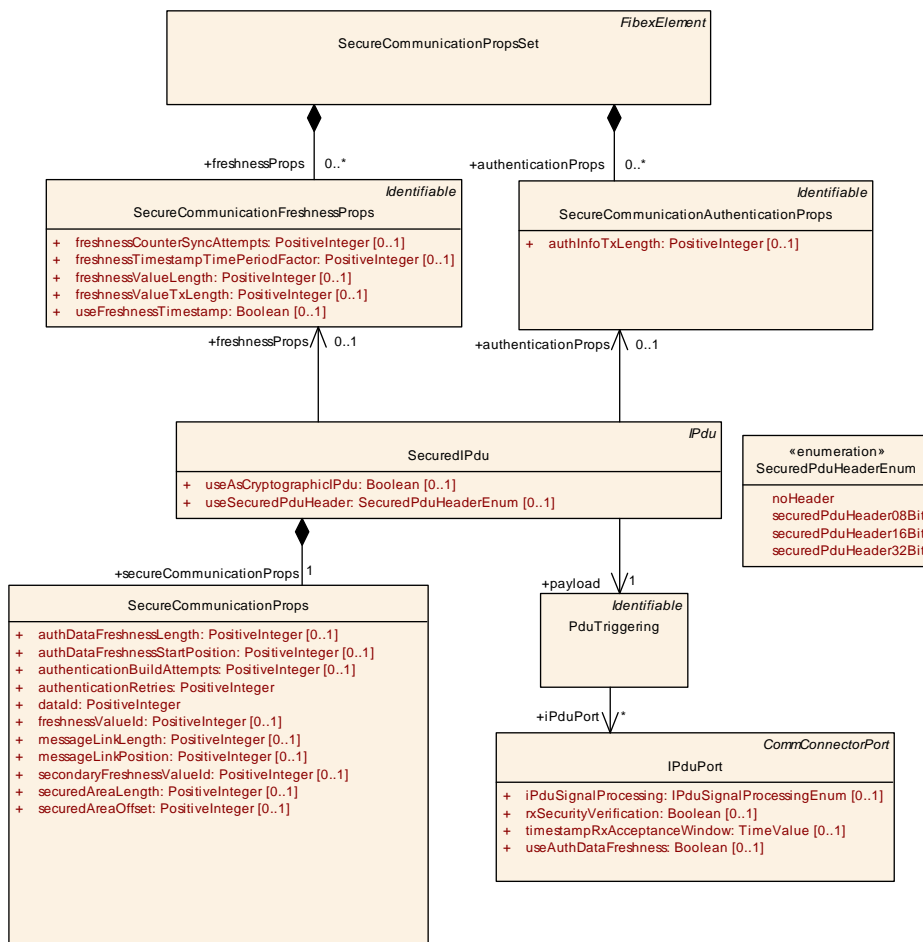


Figure 6.16: SecuredIPdu with SecureCommunicationProps

Class	SecuredIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator).</p> <p>If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu.</p> <p>Tags:atp.recommendedPackage=Pdus</p>			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable			
Attribute	Type	Mult.	Kind	Note
authentication Props	SecureCommunicationAuthenticationProps	0..1	ref	Reference to authentication properties that are valid for this SecuredIPdu.
freshnessProps	SecureCommunicationFreshnessProps	0..1	ref	Reference to freshness properties that are valid for this SecuredIPdu.
payload	PduTriggering	1	ref	Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.
secure Communication Props	SecureCommunicationProps	1	aggr	Specific configuration properties for this SecuredIPdu.
useAs Cryptographic IPdu	Boolean	0..1	attr	<p>If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.</p> <p>If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.</p>
useSecuredPdu Header	SecuredPduHeaderEnum	0..1	attr	This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

Table 6.48: SecuredIPdu

Enumeration	SecuredPduHeaderEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines the header which will be inserted into the SecuredIPdu.
Literal	Description
noHeader	<p>No header included in the SecuredPdu.</p> <p>Tags:atp.EnumerationLiteralIndex=0</p>
securedPdu Header08Bit	<p>8 Bit Secured I-PDU Header included in the Secured I-PDU.</p> <p>Tags:atp.EnumerationLiteralIndex=1</p>
securedPdu Header16Bit	<p>16 Bit Secured I-PDU Header included in the Secured I-PDU.</p> <p>Tags:atp.EnumerationLiteralIndex=2</p>
securedPdu Header32Bit	<p>32 Bit Secured I-PDU Header included in the Secured I-PDU.</p> <p>Tags:atp.EnumerationLiteralIndex=3</p>

Table 6.49: SecuredPduHeaderEnum

Class	SecureCommunicationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This meta-class contains configuration settings that are specific for an individual SecuredIPdu.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
authData Freshness Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentic PDU data that is passed to the SWC that verifies and generates the Freshness.
authData FreshnessStart Position	PositiveInteger	0..1	attr	This value determines the start position in bits of the Authentic PDU that shall be passed on to the SWC that verifies and generates the Freshness.. The bit position starts counting from the MSB of the first byte of the PDU.
authentication BuildAttempts	PositiveInteger	0..1	attr	This attribute specifies the number of authentication build attempts.
authentication Retries	PositiveInteger	1	attr	This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given SecuredIPdu. If zero is set than only one authentication attempt is done.
dataId	PositiveInteger	1	attr	This attribute defines a numerical identifier for the Secured I-PDU.
freshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.
messageLink Length	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.
messageLink Position	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.
secondary FreshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Secondary Freshness Value. The Secondary Freshness Value might be a normal counter or a time value. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
securedArea Length	PositiveInteger	0..1	attr	This attribute defines the length in bytes of the area within the payload Pdu which will be secured.
securedArea Offset	PositiveInteger	0..1	attr	This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.

Table 6.50: SecureCommunicationProps

[SecureCommunicationProps.freshnessValueId](#) does not need to be defined in case of a time-based Freshness Value. In case of a counter-based Freshness Value the [freshnessValueId](#) may be defined locally in the Ecuc or may be provided by the OEM in a System Description (this may be useful if several ECUs need to sync the freshness for a certain [freshnessValueId](#)).

Class	SecureCommunicationPropsSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Collection of properties used to configure SecuredIPdus. Tags: atp.recommendedPackage=SecureCommunicationPropsSet			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
authentication Props	SecureCommunicationAuthenticationProps	*	aggr	Authentication properties used to configure Secured IPdus.
freshnessProps	SecureCommunicationFreshnessProps	*	aggr	Freshness properties used to configure SecuredIPdus.

Table 6.51: SecureCommunicationPropsSet

Class	SecureCommunicationFreshnessProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Freshness properties used to configure SecuredIPdus.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
freshness CounterSync Attempts	PositiveInteger	0..1	attr	This attribute defines the number of Freshness Counter re-synchronization attempts when a verification failed for a Secured I-PDU. If the value is zero, there will be no additional verification attempt to synchronize with a potentially better fitting Freshness Counter value. This attribute is only applicable if useFreshnessTimestamp is FALSE.
freshness TimestampTime PeriodFactor	PositiveInteger	0..1	attr	This attribute defines a factor that specifies the time period for the Freshness Timestamp. It holds a multiplication factor that specifies the concrete meaning of a Freshness Timestamp increment by one on basis of microseconds.
freshnessValue Length	PositiveInteger	0..1	attr	This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.
freshnessValue TxLength	PositiveInteger	0..1	attr	This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.
useFreshness Timestamp	Boolean	0..1	attr	This attribute specifies whether the Freshness Value is generated through individual Freshness Counters or by a Timestamps. The value is set to TRUE when Timestamps are used.

Table 6.52: SecureCommunicationFreshnessProps

Class	SecureCommunicationAuthenticationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Authentication properties used to configure SecuredIPdus.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
authInfoTxLength	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.

Table 6.53: SecureCommunicationAuthenticationProps

[constr_3136] Allowed payload of SecuredIPdus [SecuredIPdus are allowed to reference PduTriggerings of ISignalIPdus, ContainerIPdus, DcmIPdus, MultiplexedIPdus, GeneralPurposeIPdus with category SOMEIP_SEGMENTED_IPDU and UserDefinedIPdus.]()

Please note that it is currently not supported to secure multiplexed part IPdus individually because it is not possible to reference a SecuredIPdu via MultiplexedIPdu.dynamicPart.dynamicPartAlternative or MultiplexedIPdu.staticPart in the role iPdu.

[TPS_SYST_02171] Secured Area in payload Pdu [The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete payload Pdu is secured.]([RS_SYST_00054](#))

[constr_3399] Existence of securedAreaOffset and securedAreaLength [If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.]()

[TPS_SYST_02152] Security profile [The Security profile is defined by SecureCommunicationFreshnessProps.category and by SecureCommunicationAuthenticationProps.category.]([RS_SYST_00054](#))

[constr_3324] Category of SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps [SecureCommunicationFreshnessProps that is referenced by a SecuredIPdu in the role freshnessProps shall have the same category value as the SecureCommunicationAuthenticationProps that is referenced by the same SecuredIPdu in the role authenticationProps.]()

[TPS_SYST_02153] Standardized values for the attribute category of meta-class SecureCommunicationFreshnessProps [The following values of the attribute category of meta-class SecureCommunicationFreshnessProps are reserved by the AUTOSAR standard: PROFILE_01, PROFILE_02, PROFILE_03.]([RS_SYST_00054](#))

[TPS_SYST_02154] Standardized values for the attribute category of meta-class SecureCommunicationAuthenticationProps [The following values of the attribute category of meta-class SecureCommunicationAuthenticationProps

are reserved by the AUTOSAR standard: PROFILE_01, PROFILE_02, PROFILE_03.] (*RS_SYST_00054*)

[constr_3325] SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute values for predefined categories [Table Table 6.54 defines applicable attribute values for security profiles that are standardized by AUTOSAR.]()

In other words if you want to define a SecuredIPdu in a particular Profile you have to reference the SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps and the PduTriggering of this SecuredIPdu that shall reference a SecOcCryptoServiceMapping that points to CryptoServicePrimitive that contributes the authentication Algorithm and the CryptoServiceKey that contributes the length.

Attributes	PROFILE_01	PROFILE_02	PROFILE_03
algorithmFamily	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_AES
algorithmMode	CRYPTO_ALGOMODE_CMAC	CRYPTO_ALGOMODE_CMAC	CRYPTO_ALGOMODE_CMAC
length	128 bits	128 bits	128 bits
authInfoTxLength	24 bits	24 bits	28 bits
freshnessValueLength	Not specified	0 bits	64 bits
freshnessValueTxLength	8 bits	0 bits	4 bits

Table 6.54: Security Profiles that are standardized by AUTOSAR

[constr_3339] Relation between authDataFreshnessStartPosition, authDataFreshnessLength and useAuthDataFreshness [If authDataFreshnessStartPosition and authDataFreshnessLength are set to a value for a SecuredIPdu then the useAuthDataFreshness shall be set as well to a value on all IPduPorts with communicationDirection = in that are referenced by a PduTriggering of the SecuredIPdu.]()

[TPS_SYST_02189] Setting of useSecuredPduHeader attribute [The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted directly over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).] (*RS_SYST_00054*)

In case the SecuredIPdu is contained in a ContainerIPdu or is a TP-N-SDU, its length is correctly passed by the lower layer. In these cases the SecuredIPduHeader is not needed.

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

[constr_3406] All signals before authDataFreshnessStartPosition shall have a static length [In case that

- an `ISignalIPdu` is referenced by the `SecuredIPdu` with the `payload` reference via the `PduTriggering` and
- the `authDataFreshnessStartPosition` and `authDataFreshnessLength` define the area in the `ISignalIPdu` that is taken to verify and generate the Freshness then

all `ISignals` that are mapped into the `ISignalIPdu` in front of the configured `authDataFreshnessStartPosition` shall have a static length. `]()`

Please note that parts of the Authentic `IPdu` can be used as freshness when `authDataFreshnessStartPosition` and `authDataFreshnessLength` are defined. But therefore the part of the Authentic `IPdu` to be used as the freshness has to be always available at same position in the Authentic `IPdu`.

[constr_3407] Freshness Value in Authentic `IPdu` is not allowed to be used in case of `ContainerIPdu` with a dynamic layout [If a `ContainerIPdu` that is referenced by the `SecuredIPdu` with the `payload` reference via the `PduTriggering` contains a dynamic layout (i.e. `ContainerIPdu.headerType` is set to `longHeader` or `shortHeader`) and multiple contained `IPdus` then each `IPduPort` that is referenced by the `PduTriggering` of the `SecuredIPdu` shall have the attribute `useAuthDataFreshness` set to `false`. `]()`

Please note that for `ContainerIPdus` with a dynamic layout it cannot be ensured which contained `IPdu` will be put in which position (depends on various timing and trigger conditions). Therefore `[constr_3407]` applies.

[constr_5060] Mapping of a `SecuredIPdu` into a `LinFrame` is not allowed [The mapping of a `SecuredIPdu` into a `LinFrame` with a `PduToFrameMapping` is not allowed. `]()`

In other words the usage of `Pdus` that are secured by SecOC is not allowed on a LIN Network.

6.3.2.1 Crypto Infrastructure for SecuredIPdu

From the cryptographic point of view, the usage of `SecuredIPdu` is connected to the application of two cryptographic operations, `MacGenerate` (for sending the `SecuredIPdu`) and `MacVerify` (for receiving the `SecuredIPdu`).

There are use cases for OEMs to already provide a pre-configuration of the crypto stack specifically for the handling of `SecuredIPdu`.

In order to support these use cases model elements in the meta-model are defined that allow for modeling a “crypto infrastructure” that facilitates the derivation of the configuration of operations `MacGenerate` and `MacVerify` in the crypto stack.

The formalization of a `Pdu` that is sent or received is based on the `PduTriggering` and its reference to an `IPduPort`, specifically by interpreting the attribute `IPduPort`.

`communicationDirection`. The details of the approach are summarized in Figure 6.17.

[TPS_SYST_05020] Semantics of `CryptoServiceMapping` [Meta-class `CryptoServiceMapping` represents an abstract base class for the creation of mappings in the context of cryptographic operations. Concrete sub-classes define the mapping with respect to specific cryptographic use cases, e.g. SecOC, TLS.]([RS_SYST_00054](#))

Class	<code>CryptoServiceMapping</code> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents an abstract base class for specializations of crypto service mappings.			
Base	<code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Subclasses	<code>SecOcCryptoServiceMapping</code> , <code>TlsCryptoServiceMapping</code>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.55: `CryptoServiceMapping`

[TPS_SYST_05021] Semantics of `SecOcCryptoServiceMapping` [Meta-class `SecOcCryptoServiceMapping` represents the mapping functionality required for the configuration of PDU-based secure communication by means of the `SecOc`.

In particular, the `SecOcCryptoServiceMapping` associates a `CryptoServicePrimitive` with the applicable `CryptoServiceKey`.]([RS_SYST_00054](#))

Class	<code>SecOcCryptoServiceMapping</code>			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto service mapping for the Pdu-based communication via SecOC. Tags: atp.recommendedPackage=CryptoServiceMappings			
Base	<code>ARObject</code> , <code>CryptoServiceMapping</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
authentication	<code>CryptoServicePrimitive</code>	0..1	ref	This reference identifies the applicable crypto primitive for the authentication.
cryptoService Key	<code>CryptoServiceKey</code>	0..1	ref	This reference identifies the applicable crypto key.
cryptoService Queue	<code>CryptoServiceQueue</code>	0..1	ref	This reference identifies the <code>CryptoServiceQueue</code> the processing of this <code>SecOcCryptoServiceMapping</code> shall be performed in.

Table 6.56: `SecOcCryptoServiceMapping`

[TPS_SYST_05022] Semantics of `PduTriggering.secOcCryptoMapping` [The reference `PduTriggering.secOcCryptoMapping` allows for modeling the relation of the usages (send, receive) of a `SecuredIPdu` to a given `CryptoServiceMapping` and thereby distinguish between the configuration of cryptographic operations `MacGenerate` and `MacVerify`.]([RS_SYST_00054](#))

In other words, the cryptographic use case is connected to the value of attribute `communicationDirection` of the `IPduPort` that is referenced by an `PduTriggering` that also references a `SecOcCryptoServiceMapping`:

- If the value of `communicationDirection` is set to `in` then the cryptographic use case `MacVerify` applies.
- If the value of `communicationDirection` is set to `out` then the cryptographic use case `MacGenerate` applies.

[constr_1669] Existence of `PduTriggering.secOcCryptoMapping` [The reference `PduTriggering.secOcCryptoMapping` shall only exist if the `PduTriggering` also references a `SecuredIPdu` in the role `iPdu`.]()

As the `SecOcCryptoServiceMapping` is referenced by a `PduTriggering` the `SecOcCryptoServiceMapping` needs to work for both the sender and the receivers of the corresponding Pdu in the context of a `System` of category `SYSTEM_DESCRIPTION/SYSTEM_EXTRACT` as well as in the context of a `System` of category `ECU_EXTRACT`.

[TPS_SYST_05023] Semantics of `CryptoServicePrimitive` [Meta-class `CryptoServicePrimitive` allows for the description of the applicable cryptographic algorithm.] (*RS_SYST_00054*)

Class	CryptoServicePrimitive			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto primitive. Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=CryptoPrimitives			
Base	<i>ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
algorithmFamily	String	0..1	attr	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.
algorithmMode	String	0..1	attr	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.
algorithmSecondaryFamily	String	0..1	attr	This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive. The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.

Table 6.57: CryptoServicePrimitive

[TPS_SYST_05024] Semantics of `CryptoServiceKey` [Meta-class `CryptoServiceKey` allows for the description of the applicable cryptographic key. The ability to aggregate a `ValueSpecification` in the role `developmentValue` shall be used to distribute development keys to suppliers such that crypto functionality can be adequately verified during development.] (*RS_SYST_00054*)

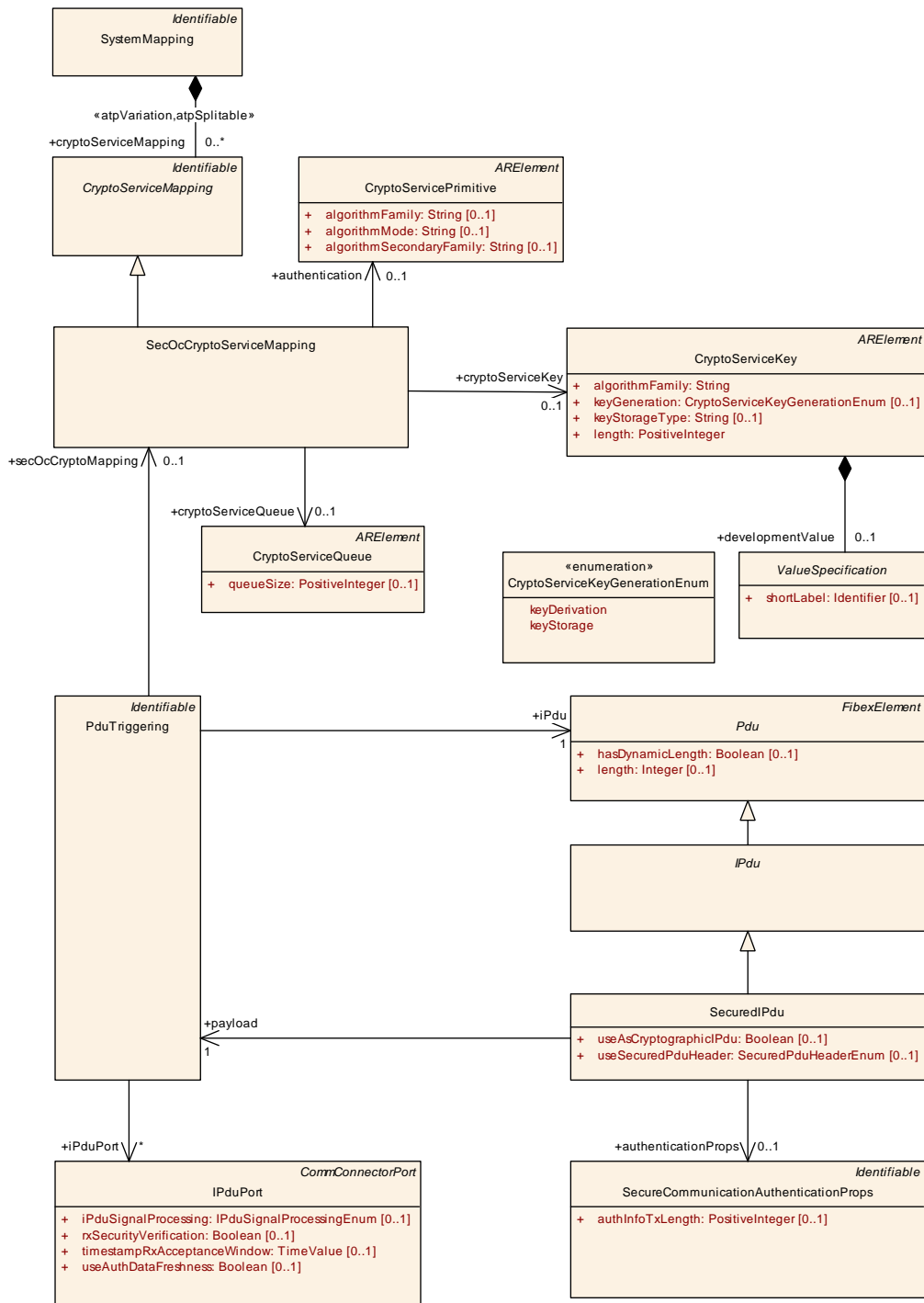


Figure 6.17: Modeling of crypto infrastructure for **SecuredIPdu**

Please note that the `developmentValue` most likely will be used in the form of a `TextValueSpecification`. However, the aggregation has still be modeled by means of using the abstract base class `ValueSpecification` to gain some headroom for future extension.

Class	CryptoServiceKey			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto key Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=CryptoDevelopmentKeys			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
algorithmFamily	String	1	attr	This attribute represent the description of the family of the applicable crypto algorithm.
development Value	ValueSpecification	0..1	aggr	This aggregation represents the ability to assign a specific value to the crypto key as part of the system description. This value can then be taken for the development of the respective ECU.
keyGeneration	CryptoServiceKeyGenerationEnum	0..1	attr	This attribute describes how a the specific cryptographic key is created.
keyStorageType	String	0..1	attr	This attribute describes where the enclosing cryptographic key shall be stored. AUTOSAR reserves specific values for this attributes but it is possible to insert custom values as well.
length	PositiveInteger	1	attr	This attribute describes the length of the cryptographic key.

Table 6.58: CryptoServiceKey

Enumeration	CryptoServiceKeyGenerationEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	This enumeration shall be taken to express the handling of a crypto key in terms of whether it is obtained from e.g. a diagnostic tester or whether it is created by derivation from a master key.
Literal	Description
keyDerivation	This means that the crypto key is created by derivation from a master key. Tags: atp.EnumerationLiteralIndex=0
keyStorage	This means that the crypto key is obtained from an external entity, e.g. a diagnostic tester. Tags: atp.EnumerationLiteralIndex=1

Table 6.59: CryptoServiceKeyGenerationEnum

[TPS_SYST_05025] Standardized values of [CryptoServicePrimitive.algorithmFamily](#) and [CryptoServiceKey.algorithmFamily](#) [The following values of attributes [CryptoServicePrimitive.algorithmFamily](#) and [CryptoServiceKey.algorithmFamily](#) are standardized by AUTOSAR:

- **CRYPTO_ALGOFAM_SHA1**: SHA1 hash
- **CRYPTO_ALGOFAM_SHA2_224**: SHA2-224 hash
- **CRYPTO_ALGOFAM_SHA2_256**: SHA2-256 hash
- **CRYPTO_ALGOFAM_SHA2_384**: SHA2-384 hash
- **CRYPTO_ALGOFAM_SHA2_512**: SHA2-512 hash
- **CRYPTO_ALGOFAM_SHA2_512_224**: SHA2-512/224 hash

- CRYPTO_ALGOFAM_SHA2_512_256: SHA2-512/256 hash
- CRYPTO_ALGOFAM_SHA3_224: SHA3-224 hash
- CRYPTO_ALGOFAM_SHA3_256: SHA3-256 hash
- CRYPTO_ALGOFAM_SHA3_384: SHA3-384 hash
- CRYPTO_ALGOFAM_SHA3_512: SHA3-512 hash
- CRYPTO_ALGOFAM_SHAKE128: SHAKE128 hash
- CRYPTO_ALGOFAM_SHAKE256: SHAKE256 hash
- CRYPTO_ALGOFAM_RIPEMD160: RIPEMD hash
- CRYPTO_ALGOFAM_BLAKE_1_256: BLAKE-1-256 hash
- CRYPTO_ALGOFAM_BLAKE_1_512: BLAKE-1-512 hash
- CRYPTO_ALGOFAM_BLAKE_2s_256: BLAKE-2s-256 hash
- CRYPTO_ALGOFAM_BLAKE_2s_512: BLAKE-2s-512 hash
- CRYPTO_ALGOFAM_3DES: 3DES cipher
- CRYPTO_ALGOFAM_AES: AES cipher
- CRYPTO_ALGOFAM_CHACHA: ChaCha cipher
- CRYPTO_ALGOFAM_RSA: RSA cipher
- CRYPTO_ALGOFAM_ED25519: ED22518 elliptic curve
- CRYPTO_ALGOFAM_BRAINPOOL: Brainpool elliptic curve
- CRYPTO_ALGOFAM_ECCNIST: NIST ECC elliptic curves
- CRYPTO_ALGOFAM_SECURECOUNTER: Secure Counter
- CRYPTO_ALGOFAM_RNG: Random Number Generator
- CRYPTO_ALGOFAM_SIPHASH: SipHash
- CRYPTO_ALGOFAM_ECIES: ECIES Cipher

](RS_SYST_00054)

[TPS_SYST_05026] Relation of `CryptoServicePrimitive.algorithmFamily` to `CryptoServiceKey.algorithmFamily` [The attribute `CryptoServiceKey.algorithmFamily` shall be taken to check with the value of `CryptoServicePrimitive.algorithmFamily` in order to make sure that the crypto key fits to the intended usage.](RS_SYST_00054)

[TPS_SYST_05027] Standardized values of `CryptoServicePrimitive.algorithmMode` [The following values of attributes `CryptoServicePrimitive.algorithmMode` are standardized by AUTOSAR:

- **CRYPTO_ALGOMODE_ECB**: Blockmode - Electronic Code Book
- **CRYPTO_ALGOMODE_CBC**: Blockmode - Cipher Block Chaining
- **CRYPTO_ALGOMODE_CFB**: Blockmode - Cipher Feedback Mode
- **CRYPTO_ALGOMODE_OFB**: Blockmode - Output Feedback Mode
- **CRYPTO_ALGOMODE_CTR**: Blockmode - Counter Modex
- **CRYPTO_ALGOMODE_GCM**: Blockmode - Galois/Counter Mode
- **CRYPTO_ALGOMODE_XTS**: XOR-encryption-based tweaked-codebook mode with ciphertext stealing
- **CRYPTO_ALGOMODE_RSAES_OAEP**: RSA Optimal Asymmetric Encryption Padding
- **CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5**: RSA encryption/decryption with PKCS#1 v1.5 padding
- **CRYPTO_ALGOMODE_RSASSA_PSS**: RSA Probabilistic Signature
- **CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5**: RSA signature with PKCS#1 v1.5
- **CRYPTO_ALGOMODE_8ROUNDS**: 8 rounds (e.g. ChaCha8)
- **CRYPTO_ALGOMODE_12ROUNDS**: 12 rounds (e.g. ChaCha12)
- **CRYPTO_ALGOMODE_20ROUNDS**: 20 rounds (e.g. ChaCha20)
- **CRYPTO_ALGOMODE_HMAC**: Hashed-based MAC
- **CRYPTO_ALGOMODE_CMAC**: Cipher-based MAC
- **CRYPTO_ALGOMODE_GMAC**: Galois MAC
- **CRYPTO_ALGOMODE_CTRDRBG**: Counter-based Deterministic Random Bit Generator
- **CRYPTO_ALGOMODE_SIPHASH_2_4**: Siphash-2-4
- **CRYPTO_ALGOMODE_SIPHASH_4_8**: Siphash-4-8

]([RS_SYST_00054](#))

Please note that it is positively supported to define custom values for attributes `CryptoServicePrimitive.algorithmFamily` and `CryptoServicePrimitive.algorithmMode` provided that the custom values are guaranteed to not clash with future extension of the AUTOSAR standard. For example, this could be achieved by using a prefix or suffix that is specific to the organization that defines the custom value.

[TPS_SYST_05028] Semantics of `CryptoServiceKey.keyStorageType`
[Attribute `CryptoServiceKey.keyStorageType` describes where the actual key shall be stored on the ECU. This attribute has been deliberately modeled as a `String` to allow future (and custom) extensions of the range of possible values.

AUTOSAR reserves the following values for this attribute:

- SHE
- RAM
- HSM
- NVM

](RS_SYST_00054)

Please note that custom values for attribute `CryptoServiceKey.keyStorageType` are supported as long as the actual values are defined in a way such that a possible clash with a later standardization by AUTOSAR becomes impossible.

The best way to achieve this is to use the company name (e.g. as a prefix) within the custom value of `CryptoServiceKey.keyStorageType`.

Class	CryptoServiceQueue			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto queue. Tags: atp.recommendedPackage=CryptoServiceQueues			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
queueSize	PositiveInteger	0..1	attr	Defines the queue size of the CryptoServiceQueue.

Table 6.60: CryptoServiceQueue

[constr_5058] Value range for `CryptoServiceQueue.queueSize` [If the `CryptoServiceQueue.queueSize` is defined it shall have a value which is equal or greater than 1.]()

6.3.3 EndToEndProtection for ISignalGroups

[TPS_SYST_01070] E2E Protection of `ISignalGroups` [It is possible to protect the inter-ECU data exchange of safety-related `ISignalGroups` which are mapped into `ISignalIPdus` using protection mechanisms provided by E2E Library.]([RS_SYST_00028](#))

[TPS_SYST_01071] E2E Protection of several `ISignalGroups` in one `ISignalIPdu` [It is possible to protect several `ISignalGroups` in one `ISignalIPdu` using several `EndToEndProtectionISignalIPdu` elements.]([RS_SYST_00028](#))

The `EndToEndProtectionISignalIPdu` element refers to the `ISignalGroup` that is to be protected and to the `ISignalIPdu` that transmits the protected `ISignalGroup`. The `dataOffset` in the `EndToEndProtectionISignalIPdu` element defines the starting position of the Array representation of the `ISignalGroup`.

The information how the referenced `ISignalGroup` shall be protected (through which E2E Profile and with which E2E settings) is defined in the `EndToEndDescription` element.

[TPS_SYST_01072] Offset attributes of `EndToEndDescription` [All offset attributes of `EndToEndDescription` are relative to the `dataOffset` with respect to the `ISignalIPdu` (absolute position of the CRC = `dataOffset` + `crcOffset`).] (*RS_SYST_00028*)

For more details, see End to End Library [20].

[TPS_SYST_01073] E2E Protection via COM Callouts [If the E2E Protection is done via COM Callouts then the `EndToEndProtectionISignalIPdu` shall be defined.] (*RS_SYST_00028*)

[TPS_SYST_01074] E2E Protection in the E2E Wrapper [If the E2E Protection is done in the E2E Wrapper then both `EndToEndProtectionISignalIPdu` and `EndToEndProtectionVariablePrototype` shall be defined.] (*RS_SYST_00028*)

For more details, see Software Component Template specification [5].

[constr_1002] End-to-end protection does not support n:1 communication [As the n:1 communication scenario implies that probably not all senders use the same `dataId` this scenario is explicitly not supported.] ()

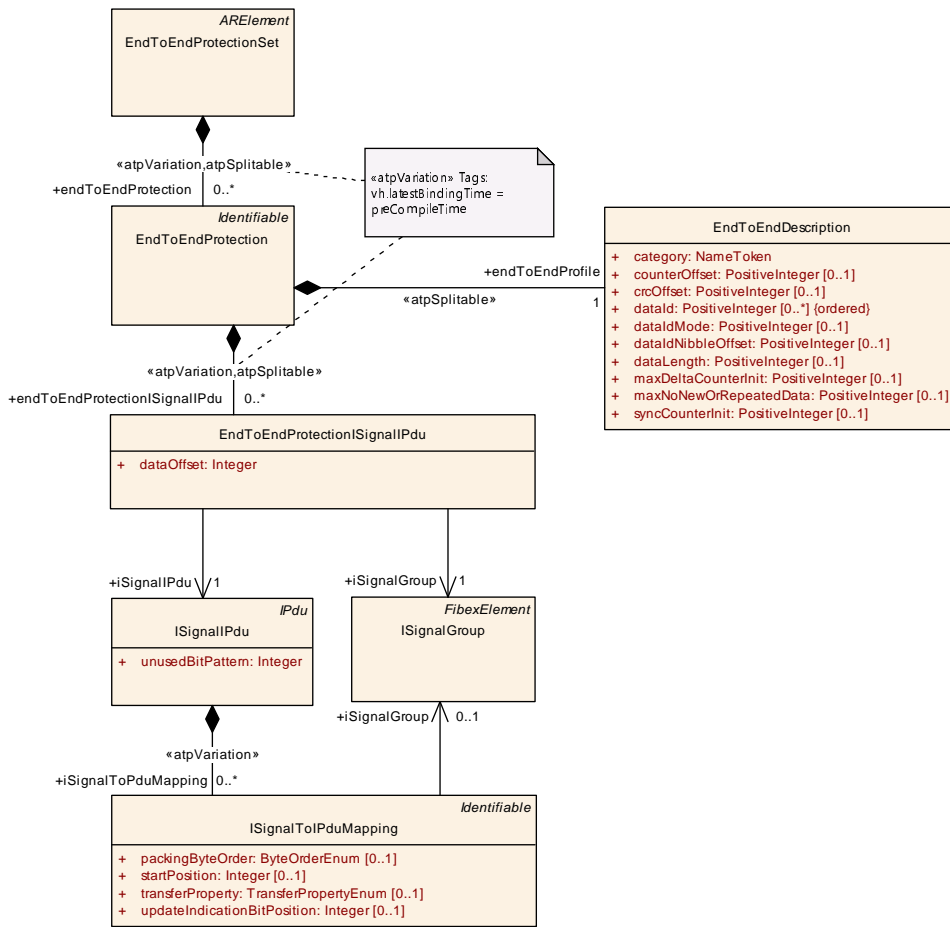


Figure 6.18: EndToEndProtection for COM IPdus

Class	EndToEndProtectionSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This represents a container for collection EndToEndProtectionInformation. Tags: atp.recommendedPackage=EndToEndProtectionSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
endToEndProtection	EndToEndProtection	*	aggr	This is one particular EndToEndProtection. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 6.61: EndToEndProtectionSet

Class	EndToEndProtection			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This meta-class represents the ability to describe a particular end to end protection.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
endToEndProfile	EndToEndDescription	1	aggr	This represents the particular EndToEndDescription. Stereotypes: atpSplitable Tags: atp.Splitkey=endToEndProfile
endToEndProtectionISignalPdu	EndToEndProtectionISignalPdu	*	aggr	Defines to which ISignalPdu - ISignalGroup pair this EndToEndProtection shall apply. In case several ISignalGroups are used to transport the data (e.g. fan-out in the RTE) there may exist several EndToEndProtectionISignalPdu definitions. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=variationPoint.shortLabel vh.latestBindingTime=preCompileTime
endToEndProtectionVariablePrototype	EndToEndProtectionVariablePrototype	*	aggr	Defines to which VariableDataPrototypes in the roles of one sender and one or more receivers this EndToEndProtection applies. It shall be possible to aggregate several EndToEndProtectionVariablePrototype in case additional hierarchical decompositions are introduced subsequently. In this case one particular PortPrototype is split into multiple PortPrototypes and connectors, all representing the same data entity. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortLabel, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 6.62: EndToEndProtection

Class	EndToEndProtectionISignalPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::EndToEndProtection			
Note	It is possible to protect the inter-ECU data exchange of safety-related ISignalGroups at the level of COM IPdus using protection mechanisms provided by E2E Library. For each ISignalGroup to be protected, a separate EndToEndProtectionISignalPdu element shall be created within the EndToEndProtectionSet. The EndToEndProtectionISignalPdu element refers to the ISignalGroup that is to be protected and to the ISignalPdu that transmits the protected ISignalGroup. The information how the referenced ISignalGroup shall be protected (through which E2E Profile and with which E2E settings) is defined in the EndToEndDescription element.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dataOffset	Integer	1	attr	This attribute defines the beginning offset (in bits) of the Array representation of the Signal Group (including CRC, counter and application signal group) in the IPdu. This attribute is mandatory and the dataOffset shall always be defined.
iSignalGroup	ISignalGroup	1	ref	Reference to the ISignalGroup that is to be protected.





Class	EndToEndProtectionISignalPdu			
iSignalPdu	ISignalPdu	1	ref	Reference to the ISignalPdu that transmits the protected ISignalGroup.

Table 6.63: EndToEndProtectionISignalPdu

Class	EndToEndDescription			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This meta-class contains information about end-to-end protection. The set of applicable attributes depends on the actual value of the category attribute of EndToEndProtection.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
category	NameToken	1	attr	The category represents the identification of the concrete E2E profile. The applicable values are specified in a semantic constraint and determine the applicable attributes of EndToEndDescription. Tags: xml.sequenceOffset=-100
counterOffset	PositiveInteger	0..1	attr	Bit offset of Counter from the beginning of the Array representation of the Signal Group/VariableDataPrototype (MSB order, bit numbering: bit 0 is the least important). The offset shall be a multiplicity of 4 and it should be 8 whenever possible. For example, offset 8 means that the counter will take the low nibble of the byte 1, i.e. bits 8 .. 11. If counterOffset is not present the value is defined by the selected profile. Tags: xml.sequenceOffset=-50
crcOffset	PositiveInteger	0..1	attr	Bit offset of CRC from the beginning of the Array representation of the Signal Group/VariableDataPrototype (MSB order, bit numbering: bit 0 is the least important). The offset shall be a multiplicity of 8 and it should be 0 whenever possible. For example, offset 8 means that the CRC will take the byte 1, i.e. bits 8..15. If crcOffset is not present the value is defined by the selected profile. Tags: xml.sequenceOffset=-60
dataId (ordered)	PositiveInteger	*	attr	This represents a unique numerical identifier. Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection. Tags: xml.sequenceOffset=-90
dataIdMode	PositiveInteger	0..1	attr	There are three inclusion modes how the implicit two-byte Data ID is included in the one-byte CRC: <ul style="list-style-type: none"> • dataIdMode = 0: Two bytes are included in the CRC (double ID configuration) This is used in variant 1A. • dataIdMode = 1: One of the two bytes byte is included, alternating high and low byte, depending on parity of the counter (alternating ID configuration). For even counter low byte is





Class	EndToEndDescription			
				<p>△ included; For odd counters the high byte is included. This is used in variant 1B.</p> <ul style="list-style-type: none"> dataIDMode = 2: Only low byte is included, high byte is never used. This is applicable if the IDs in a particular system are 8 bits. dataIDMode = 3: The low byte is included in the implicit CRC calculation, the low nibble of the high byte is transmitted along with the data (i.e. it is explicitly included), the high nibble of the high byte is not used. This is applicable for the IDs up to 12 bits. <p>Tags:xml.sequenceOffset=-85</p>
dataIDNibbleOffset	PositiveInteger	0..1	attr	<p>Bit offset of the low nibble of the high byte of Data ID. The applicability of this attribute is controlled by [constr_1261].</p> <p>Tags:xml.sequenceOffset=-25</p>
dataLength	PositiveInteger	0..1	attr	<p>This attribute represents the length of the Array representation of the Signal Group/VariableDataPrototype including CRC and Counter in bits.</p> <p>Tags:xml.sequenceOffset=-80</p>
maxDeltaCounterInit	PositiveInteger	0..1	attr	<p>Initial maximum allowed gap between two counter values of two consecutively received valid Data, i.e. how many subsequent lost data is accepted. For example, if the receiver gets Data with counter 1 and MaxDeltaCounter Init is 1, then at the next reception the receiver can accept Counters with values 2 and 3, but not 4.</p> <p>Note that if the receiver does not receive new Data at a consecutive read, then the receiver increments the tolerance by 1.</p> <p>Tags:xml.sequenceOffset=-70</p>
maxNoNewOrRepeatedData	PositiveInteger	0..1	attr	<p>The maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.</p> <p>Tags:xml.sequenceOffset=-40</p>
syncCounterInit	PositiveInteger	0..1	attr	<p>Number of Data required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.</p> <p>Tags:xml.sequenceOffset=-30</p>

Table 6.64: EndToEndDescription

The `maxDeltaCounterInit`, `maxNoNewOrRepeatedData` and `syncCounterInit` values can also be specified in the `ReceiverComSpec`. This allows the definition of receiver specific values. Values for `maxDeltaCounterInit`, `maxNoNewOrRepeatedData` and `syncCounterInit` that are defined in the `ReceiverComSpec` override the possible values in the `EndToEndDescription` class. More details can be found in the Software Component Template specification [5].

The supported E2E profiles (possible values of category in `EndToEndDescription`) are described in the Software Component Template [5] and the End to End Library [20].

6.3.4 GeneralPurposeConnection

In some cases it is important to describe a relation between different `PduTriggerings` that are defined on the same `PhysicalChannel`, e.g. to create a link between a Rx-Pdu and a Tx-Pdu. The `GeneralPurposeConnection` meta-class is able to reference a number of `PduTriggerings` and thereby to set the referenced `PduTriggerings` into a relationship that is defined by the `GeneralPurposeConnection`.

[TPS_SYST_02170] category of the GeneralPurposeConnection [The category of the `GeneralPurposeConnection` is used to define the purpose of the relationship between the referenced `PduTriggerings`.]()

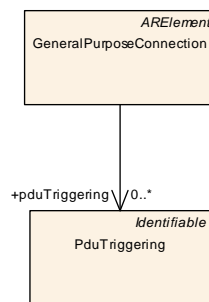


Figure 6.19: GeneralPurposeConnection

Class	GeneralPurposeConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::GeneralPurposeConnection			
Note	This meta-class allows to describe the relationship between several <code>PduTriggerings</code> that are defined on the same <code>PhysicalChannel</code> , e.g. to create a link between Rx and Tx Pdu that are used for request/response. Tags: atp.recommendedPackage=GeneralPurposeConnections			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
pduTriggering	PduTriggering	*	ref	Reference to <code>PduTriggerings</code> that are connected to each other by a <code>GeneralPurposeConnection</code> .

Table 6.65: GeneralPurposeConnection

[constr_3384] PduTriggerings referenced by GeneralPurposeConnection shall be defined on the same PhysicalChannel [The `PduTriggerings` that are referenced by the `GeneralPurposeConnection` in the role `pduTriggering` shall be defined on the same `PhysicalChannel`.]()

[constr_3383] Standardized values for the attribute category of meta-class GeneralPurposeConnection [The following values of the attribute `category` of meta-class `GeneralPurposeConnection` are reserved by the AUTOSAR standard:

- XcpChannel

]()

The XcpChannel creates a link between one Tx-Pdu and one Rx-Pdu that are used for request/response from one master.

[constr_3385] XcpChannel is allowed to reference exactly two PduTriggerings [In case that the `category` of meta-class `GeneralPurposeConnection` is set to the value `XcpChannel` the `GeneralPurposeConnection` is allowed to reference exactly two `PduTriggerings` in the role `pduTriggering`.]()

[constr_3386] XcpChannel is only allowed to reference PduTriggerings of GeneralPurposeIPdus with category XCP [In case that the `category` of meta-class `GeneralPurposeConnection` is set to the value `XcpChannel` the `GeneralPurposeConnection` is allowed to reference `PduTriggerings` of `GeneralPurposeIPdus` with category `XCP`.]()

6.4 IPdu Timing

AUTOSAR COM allows configuring statically two different transmission modes for each IPdu (True and False). `TransmissionModeDeclaration` uses a transmission mode selector, calculated from a number of individual `TransmissionModeConditions` or `ModeDrivenTransmissionModeConditions` to decide which of the two modes is selected. It is possible to switch between the transmission modes during runtime.

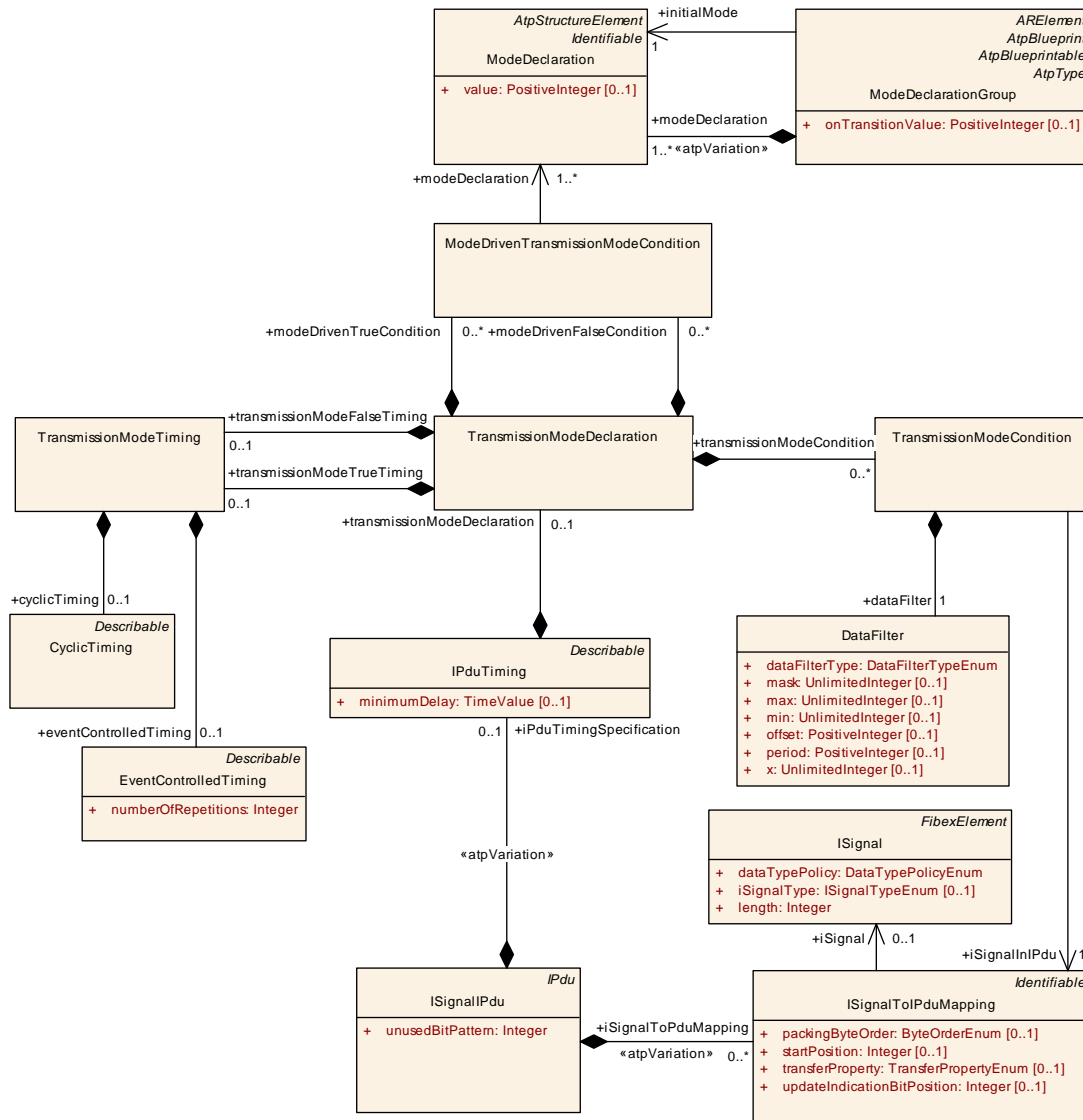


Figure 6.20: IPdu Timing

[TPS_SYST_01075] Signal content evaluation via [TransmissionModeCondition](#) [The signal content can be evaluated as the transmission mode selector via the [TransmissionModeConditions](#).] ([RS_SYST_00037](#))

[TPS_SYST_01076] Mode evaluation via [modeDrivenTrueCondition](#) or [modeDrivenFalseCondition](#) [Mode conditions can be evaluated as the transmission mode selector via the [modeDrivenTrueConditions](#) or [modeDrivenFalseConditions](#).] ([RS_SYST_00037](#))

[constr_3045] Signal content evaluation vs. Mode evaluation [The mode evaluation and the signal content evaluation shall not be used in the same [IPdu](#). A mix of these two types is not allowed.] ()

To use the signal content evaluation a [TransmissionModeCondition](#) can be attached to each signal within an [IPdu](#). Each [TransmissionModeCondition](#) contains a reference to a signal and to an assigned filter. The filter condition is used for

the selection of the transmission mode. If at least one condition in the signal content evaluation is true, Transmission Mode "TRUE" shall be used for this IPdu. In all other cases, the Transmission Mode "FALSE" shall be used. More details can be found in the COM Specification [21].

[constr_3046] Consistency of `TransmissionModeCondition.iSignalInIPdu`
 [The `ISignalToIPduMapping` referenced by the `TransmissionModeCondition` in the role `iSignalInIPdu` shall belong to the same `ISignalIPdu` as the `TransmissionModeCondition`.]()

In the mode driven evaluation `ModeDeclarations` are evaluated. The `transmissionModeFalseTiming` is activated if all defined `modeDrivenFalseConditions` evaluate to true and the `transmissionModeTrueTiming` is activated if all defined `modeDrivenTrueConditions` evaluate to true. Each condition that is defined by `ModeDrivenTransmissionModeCondition` evaluates to true if one of the referenced `ModeDeclarations` is active.

The `TransmissionModeDeclaration` element aggregates the `TransmissionModeTiming` in two different roles: `transmissionModeTrueTiming` and `transmissionModeFalseTiming`. The available COM Transmission Mode Timings can be described by the `CyclicTiming` and `EventControlledTiming` elements (see Table 6.66) that are aggregated by the `TransmissionModeTiming` class.

[TPS_SYST_01077] Mapping of Com Transmission Modes to System Template elements [The mapping of COM Transmission Modes to System Template elements is described in Table 6.66.] (*RS_SYST_00037*)

COM Modes	Transmission	Description	realization in System Template
Periodic		Transmissions occur indefinitely with a fixed period between them	<code>CyclicTiming</code>
Direct/n-times		Event driven transmission with n-1 repetitions	<code>EventControlledTiming</code>
Mixed		Periodic transmission with direct/n-times transmissions in between	<code>EventControlledTiming</code> and <code>CyclicTiming</code>
None		No transmission	no timing assigned

Table 6.66: COM Transmission Modes

Class	<code>TransmissionModeDeclaration</code>
Package	<code>M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing</code>





Class		TransmissionModeDeclaration		
Note	<p>AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES (True and False) for each I-PDU.</p> <p>As TransmissionMode selector the signal content can be evaluated via transmissionModeCondition (implemented directly in the COM module) or mode conditions can be defined with the modeDrivenTrueCondition or modeDrivenFalseCondition (evaluated by BswM and invoking Com_SwitchIpduTxMode COM API). If modeDrivenTrueCondition and modeDrivenFalseCondition are defined they shall never evaluate to true both at the same time.</p> <p>The mixing of Transmission Mode Switch via API and signal value is not allowed.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
modeDrivenFalseCondition	ModeDrivenTransmissionModeCondition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDrivenFalseConditions evaluate to true (AND associated) the transmissionModeFalseTiming shall be activated. modeDrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time.
modeDrivenTrueCondition	ModeDrivenTransmissionModeCondition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDrivenTrueConditions evaluate to true (AND associated) the transmissionModeTrueTiming shall be activated. modeDrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time.
transmissionModeCondition	TransmissionModeCondition	*	aggr	The Transmission Mode Selector evaluates the conditions for a subset of signals and decides which transmission mode should be used. In case only one transmission mode is used there is no need for the "TransmissionModeCondition" and its sub-structure. In case the transmission mode shall be switched using the COM-API "Com_SwitchIpduTxMode" there is no need for the "TransmissionModeCondition" and its sub-structure.
transmissionModeFalseTiming	TransmissionModeTiming	0..1	aggr	Timing Specification if the COM Transmission Mode is false. The Transmission Mode Selector is defined to be false, if all Conditions evaluate to false.
transmissionModeTrueTiming	TransmissionModeTiming	0..1	aggr	Timing Specification if the COM Transmission Mode is true. The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true.

Table 6.67: TransmissionModeDeclaration

Class		TransmissionModeCondition		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	<p>Possibility to attach a condition to each signal within an I-PDU.</p> <p>If at least one condition evaluates to true, TRANSMISSION MODE True shall be used for this I-Pdu. In all other cases, the TRANSMISSION MODE FALSE shall be used.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
dataFilter	DataFilter	1	aggr	Possibilities to define conditions
iSignalInIPdu	ISignalToIPduMapping	1	ref	Reference to a signal to which a condition is attached.

Table 6.68: TransmissionModeCondition

Class	ModeDrivenTransmissionModeCondition			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	The condition defined by this class evaluates to true if one of the referenced modeDeclarations (OR associated) is active. All referenced modeDeclarations shall be from the same ModeDeclarationGroup. The condition is used to define which TransmissionMode shall be activated using Com_SwitchIpduTx Mode.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
mode Declaration	ModeDeclaration	1..*	ref	Reference to one modeDeclaration which is OR associated in the context of the ModeDrivenTransmission ModeCondition.

Table 6.69: ModeDrivenTransmissionModeCondition

The [ModeDeclaration](#) and the [ModeDeclarationGroup](#) is described in more detail in the Software Component Template Specification [5].

Class	TransmissionModeTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: <ul style="list-style-type: none"> • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned) 			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
cyclicTiming	CyclicTiming	0..1	aggr	Periodic Transmission Mode.
eventControlled Timing	EventControlledTiming	0..1	aggr	Direct Transmission Mode.

Table 6.70: TransmissionModeTiming

6.4.1 Data Filter configuration

Data Filters are used on sender side to configure Transmission Mode Conditions (TMC). On receiver side Data Filters can be used as filtering mechanisms for signals (see `ISignalPort` element). More details about the usage of `DataFilters` can be found in the Software Component Template Specification [5].

[TPS_SYST_02013] Usage of `dataFilters` on GroupSignals on receiver side
 [If the `dataFilter` of one `GroupSignal` evaluates to false the whole `ISignal-Group` in which the `GroupSignal` is contained shall be discarded.]()

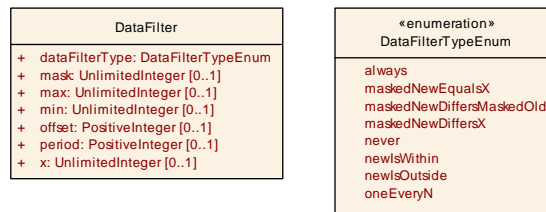


Figure 6.21: Data Filter

Class	DataFilter			
Package	M2::AUTOSARTemplates::CommonStructure::Filter			
Note	Base class for data filters. The type of the filter is specified in attribute <code>dataFilterType</code> . Some of the filter types require additional arguments which are specified as attributes of this class.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
<code>dataFilterType</code>	DataFilterTypeEnum	1	attr	This attribute specifies the type of the filter.
<code>mask</code>	UnlimitedInteger	0..1	attr	Mask for old and new value.
<code>max</code>	UnlimitedInteger	0..1	attr	Value to specify the upper boundary
<code>min</code>	UnlimitedInteger	0..1	attr	Value to specify the lower boundary
<code>offset</code>	PositiveInteger	0..1	attr	Specifies the initial number of messages to occur before the first message is passed
<code>period</code>	PositiveInteger	0..1	attr	Specifies number of messages to occur before the message is passed again
<code>x</code>	UnlimitedInteger	0..1	attr	Value to compare with

Table 6.71: DataFilter

Enumeration	DataFilterTypeEnum
Package	M2::AUTOSARTemplates::CommonStructure::Filter
Note	This enum specifies the supported <code>DataFilterTypes</code> .
Literal	Description
always	No filtering is performed so that the message always passes. Tags: atp.EnumerationLiteralIndex=0





Enumeration	DataFilterTypeEnum
maskedNewDiffers MaskedOld	<p>Pass messages where the masked value has changed.</p> $(new_value \& mask) \neq (old_value \& mask)$ new_value: current value of the message old_value: last value of the message (initialized with the initial value of the message, updated with new_value if the new message value is not filtered out) Tags: atp.EnumerationLiteralIndex=1
maskedNewDiffers X	<p>Pass messages whose masked value is not equal to a specific value x</p> $(new_value \& mask) \neq x$ new_value: current value of the message Tags: atp.EnumerationLiteralIndex=2
maskedNewEquals X	<p>Pass messages whose masked value is equal to a specific value x</p> $(new_value \& mask) == x$ new_value: current value of the message Tags: atp.EnumerationLiteralIndex=3
never	<p>The filter removes all messages.</p> Tags: atp.EnumerationLiteralIndex=4
newIsOutside	<p>Pass a message if its value is outside a predefined boundary.</p> $(min > new_value) \text{ OR } (new_value > max)$ Tags: atp.EnumerationLiteralIndex=5
newIsWithin	<p>Pass a message if its value is within a predefined boundary.</p> $min \leq new_value \leq max$ Tags: atp.EnumerationLiteralIndex=6
oneEveryN	<p>Pass a message once every N message occurrences.</p> <p>Algorithm: $occurrence \% period == offset$ Start: $occurrence = 0$.</p> <p>Each time the message is received or transmitted, occurrence is incremented by 1 after filtering. Length of occurrence is 8 bit (minimum). Tags:atp.EnumerationLiteralIndex=7 </p>

Table 6.72: DataFilterTypeEnum

6.4.2 Cyclic Timing

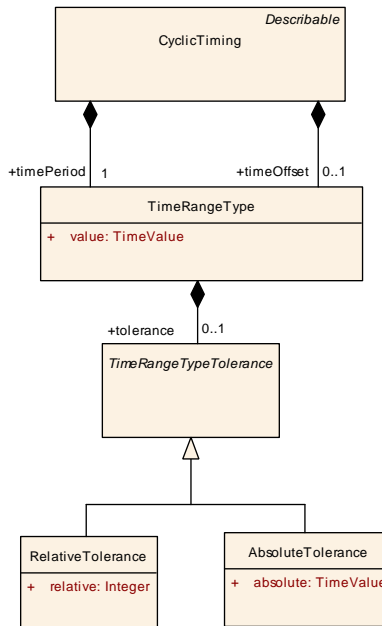


Figure 6.22: Cyclic Timing

Class	CyclicTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Specification of a cyclic sending behavior.			
Base	ARObject, Describable			
Attribute	Type	Mult.	Kind	Note
timeOffset	TimeRangeType	0..1	aggr	This attribute specifies the time until first transmission of this I-PDU. This attribute defines the time between Com_IpduGroupStart and the first transmission of the cyclic part of this transmission request for this I-PDU.
timePeriod	TimeRangeType	1	aggr	Period of the repetition of cyclic transmissions.

Table 6.73: CyclicTiming

6.4.3 EventControlled Timing

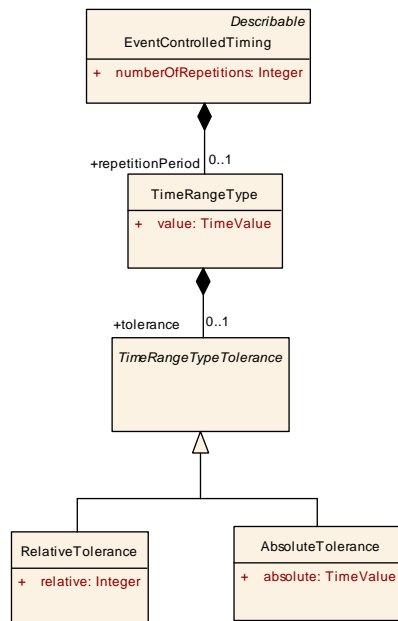


Figure 6.23: EventControlled Timing

Class	EventControlledTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Specification of a event driven sending behavior. The PDU is sent n (numberOfRepeat + 1) times separated by the repetitionPeriod. If numberOfRepeats = 0, then the Pdu is sent just once.			
Base	ARObject, <i>Describable</i>			
Attribute	Type	Mult.	Kind	Note
numberOfRepetitions	Integer	1	attr	Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.
repetitionPeriod	TimeRangeType	0..1	aggr	The repetitionPeriod specifies the time in seconds that elapses before the pdu can be sent the next time (Minimum repeat gap between two pdus). The repetition Period is optional in case that no repetitions are configured.

Table 6.74: EventControlledTiming

Class	TimeRangeType			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	The timeRange can be specified with the value attribute. Optionally a tolerance can be defined.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
tolerance	TimeRangeTypeTolerance	0..1	aggr	Optional specification of a tolerance.





Class	TimeRangeType			
value	TimeValue	1	attr	Average value of a date (in seconds)

Table 6.75: TimeRangeType

Class	RelativeTolerance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Maximum allowable deviation			
Base	<i>ARObject, TimeRangeTypeTolerance</i>			
Attribute	Type	Mult.	Kind	Note
relative	Integer	1	attr	Maximum allowable deviation in percent (percent of the corresponding TimeValue).

Table 6.76: RelativeTolerance

Class	AbsoluteTolerance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Maximum allowable deviation			
Base	<i>ARObject, TimeRangeTypeTolerance</i>			
Attribute	Type	Mult.	Kind	Note
absolute	TimeValue	1	attr	Maximum allowable deviation in duration (in seconds)

Table 6.77: AbsoluteTolerance

6.4.4 Configuration of a trigger for COM_TriggerIPduSend API call

In the AUTOSAR BswM module a `BswMAction` with `BswMTriggerIPduSend` may be defined. The COM API `Com_TriggerIPDUSe` is called when this action is configured. By the call of `Com_TriggerIPDUSe` an IPdu with a given ID is triggered for transmission.

With such a configuration a single transmission of an IPdu can be configured that is independent of the configured COM transmission modes, e.g. in case of a vehicle mode change.

In a System Description the usage of the `Com_TriggerIPDUSe` API is defined with the `TriggerIPduSendCondition` that is aggregated by the `PduTriggering` in the role `triggerIPduSendCondition`. The `TriggerIPduSendCondition` defines the trigger for the `Com_TriggerIPDUSe` API call.

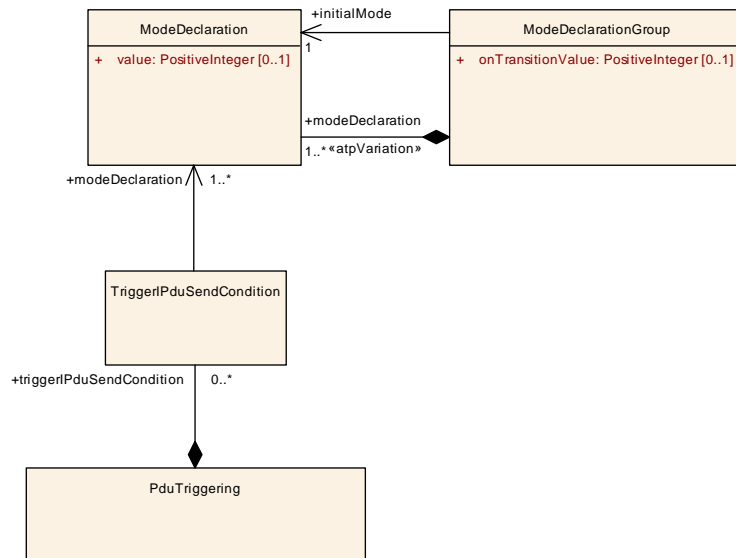


Figure 6.24: TriggerIPduSendCondition

Class	TriggerIPduSendCondition			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	The condition defined by this class evaluates to true if one of the referenced modeDeclarations (OR associated) is active. The condition is used to define when the Pdu is triggered with the Com_TriggerIPDUSe API call.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
mode Declaration	ModeDeclaration	1..*	ref	Reference to one modeDeclaration which is OR associated in the context of the TriggerIPduSend Condition.

Table 6.78: TriggerIPduSendCondition

Only if all defined `TriggerIPduSendConditions` evaluate to true (AND associated) the `Com_TriggerIPDUSe` API shall be called.

[constr_3211] PduTriggerings with triggerIPduSendCondition [Only PduTriggerings with references to ISignalIPdus are allowed to contain a triggerIPduSendCondition.]()

Please note that OR Conditions defined by the `TriggerIPduSendCondition.modeDeclaration` are evaluated first. The AND Conditions defined by `PduTriggering.triggerIPduSendConditions` are evaluated after the OR Conditions.

6.5 I-Pdu Multiplexer

Multiplexing is used to transport varying Com IPdus at the same position in a single multiplexed IPdu. A multiplexed IPdu consists of a dynamic part, a selector field and an optional static part. According to the value of the selector field the dynamic part can have a different layout.

[TPS_SYST_01078] Dynamic Part of a MultiplexedIPdu [For each alternative of a MultiplexedIPdu there is exactly one Com IPdu that is transmitted in the dynamic part.]()

[TPS_SYST_01079] Static Part of a MultiplexedIPdu [The static part of a MultiplexedIPdu is the same regardless of the selector field and consists of exactly one Com IPdu.]()

The MultiplexedIPdu element contains attributes that describe the position and the length of a selector within an IPdu. A selector is a bitfield of certain length, by the value of which the corresponding data region of the dynamic part shall be interpreted dynamically, i.e. at run-time.

[constr_3007] selectorFieldCodes for dynamic part alternatives [The selectorFieldCodes for the dynamic part alternatives within one MultiplexedIPdu shall differ from each other.]()

[constr_3097] Overlapping of segments of one MultiplexedIPdu is not allowed [The segments defined by the SegmentPosition elements of one and the same MultiplexedIPdu - aggregated via StaticPart and DynamicPart - shall not overlap.]()

[constr_3098] Defined segments of one MultiplexedIPdu shall not exceed the length of the MultiplexedIPdu [The segments defined by the SegmentPosition elements of one and the same MultiplexedIPdu - aggregated via StaticPart and DynamicPart - shall not exceed the length of the MultiplexedIPdu.]()

[constr_3099] Defined segments in a DynamicPart shall not exceed the length of any DynamicPartAlternative.iPdu [The segments defined by the SegmentPosition elements aggregated in the DynamicPart of a MultiplexedIPdu shall not exceed the length of any DynamicPartAlternative.iPdu.]()

[constr_3100] Defined segments in a StaticPart shall not exceed the length of the StaticPart.iPdu [The segments defined by the SegmentPosition elements aggregated in the StaticPart of a MultiplexedIPdu shall not exceed the length of the StaticPart.iPdu.]()

[constr_3101] Signal representation of selector field for DynamicPartAlternative [Every ISignalIPdu that is referenced by the DynamicPartAlternative shall contain an ISignal that represents the selector field. The selector field signal shall be located at the position that is described by the selectorFieldLength and selectorFieldStartPosition.]()

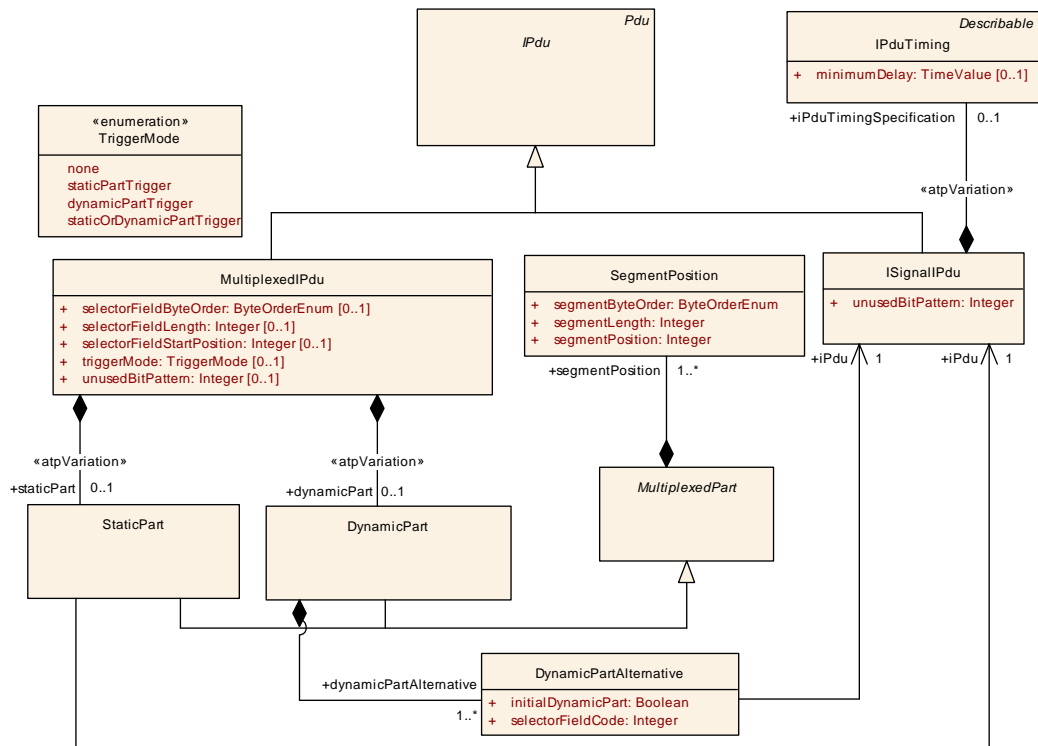


Figure 6.25: I-Pdu Multiplexer (FibexCore: IPDUMultiplexerOverview)

Enumeration	TriggerMode
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	IPduM can be configured to send a transmission request for the new multiplexed I-PDU to the PDU-Router because of conditions/ modes.
Literal	Description
dynamicPartTrigger	IPduM sends a transmission request to the PduR if a dynamic part is received. Tags: atp.EnumerationLiteralIndex=0
none	IPduM does not trigger transmission because of receiving anything of this IPdu in case of Trigger Transmit. Tags: atp.EnumerationLiteralIndex=1
staticOrDynamicPartTrigger	IPduM sends a transmission request to the PduR if a static or dynamic part is received. Tags: atp.EnumerationLiteralIndex=2
staticPartTrigger	IPduM sends a transmission request to the PduR if a static part is received. Tags: atp.EnumerationLiteralIndex=3

Table 6.79: TriggerMode

Class	MultiplexedIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	<p>A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selector Field. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.</p> <p>A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.</p> <p>Tags:atp.recommendedPackage=Pdus</p>			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable			
Attribute	Type	Mult.	Kind	Note
dynamicPart	DynamicPart	0..1	aggr	<p>According to the value of the selector field some parts of the IPdu have a different layout.</p> <p>In a complete System Description a MultiplexedIPdu shall contain a DynamicPart. The following use cases support the multiplicity to be 0..1:</p> <ul style="list-style-type: none"> • If a MultiplexedIPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedIPdu doesn't need to be described in the System Extract/Ecu Extract. • If a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu then the dynamicPart does not need to be described in the System Extract/Ecu Extract. <p>atpVariation: Content of a multiplexed PDU can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>





Class	MultiplexedIPdu			
selectorField ByteOrder	ByteOrderEnum	0..1	attr	<p>This attribute defines the order of the bytes of the selector field and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute.</p> <p>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>
selectorField Length	Integer	0..1	attr	<p>The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>
selectorField StartPosition	Integer	0..1	attr	<p>This parameter is necessary to describe the position of the selector field within the IPdu.</p> <p>Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorFieldByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>
staticPart	StaticPart	0..1	aggr	<p>The static part of the multiplexed IPdu is the same regardless of the selector field. The static part is optional.</p> <p>atpVariation: Content of a multiplexed PDU can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
triggerMode	TriggerMode	0..1	attr	<p>IPduM can be configured to send a transmission request for the new multiplexed IPdu to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration.</p> <p>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>





Class	MultiplexedIPdu			
unusedBit Pattern	Integer	0..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> <p>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>

Table 6.80: MultiplexedIPdu

Class	StaticPart			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.			
Base	ARObject, MultiplexedPart			
Attribute	Type	Mult.	Kind	Note
iPdu	ISignalIPdu	1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu.

Table 6.81: StaticPart

Class	DynamicPart			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Dynamic part of a multiplexed I-Pdu. Reserved space which is used to transport varying SignalIPdus at the same position, controlled by the corresponding selectorFieldCode.			
Base	ARObject, MultiplexedPart			
Attribute	Type	Mult.	Kind	Note
dynamicPart Alternative	DynamicPartAlternative	1..*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu.

Table 6.82: DynamicPart

Class	DynamicPartAlternative			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
initialDynamic Part	Boolean	1	attr	<p>Dynamic part that shall be used to initialize this multiplexed IPdu.</p> <p>Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.</p>





Class	DynamicPartAlternative			
iPdu	ISignallIPdu	1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu.
selectorField Code	Integer	1	attr	The selector field is part of a multiplexed IPdu. It consists of contiguous bits. The value of the selector field selects the layout of the multiplexed part of the IPdu.

Table 6.83: DynamicPartAlternative

Class	MultiplexedPart (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The StaticPart and the DynamicPart have common properties. Both can be separated in multiple segments within the multiplexed PDU.			
Base	<i>ARObject</i>			
Subclasses	DynamicPart , StaticPart			
Attribute	Type	Mult.	Kind	Note
segment Position	SegmentPosition	1..*	aggr	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. Therefore the StaticPart and the DynamicPart can contain multiple SegmentPositions.

Table 6.84: MultiplexedPart

Class	SegmentPosition			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalPdu are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalPdu are copied into this first segment and so on.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
segmentByteOrder	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.
segmentLength	Integer	1	attr	Data Length of the segment in bits.
segmentPosition	Integer	1	attr	Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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 6.85: SegmentPosition

[constr_3247] Byte order mix within a MultiplexedIPdu is not allowed [The segmentByteOrder of all SegmentPositions and the selectorFieldByteOrder shall have the same value in the MultiplexedIPdu.]()

[constr_3223] No ByteOrderEnum.opaque allowed for MultiplexedIPdu.selectorFieldByteOrder [The values of MultiplexedIPdu.selectorFieldByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.]()

[constr_3224] No ByteOrderEnum.opaque allowed for SegmentPosition.segmentByteOrder. [The values of SegmentPosition.segmentByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.]()

Figure 6.26 shows an example of an IPdu Multiplexer. The static part of the multiplexed IPdu contains ComIPduA. The value of the selector field in the dynamic part decides which content is transmitted. ComIPduB is transmitted if the selector field value is "0". ComIPduC is transmitted if the selector field value is "1".

The static and the dynamic part can consist of more than one element. These sub parts of the static or dynamic parts are called segments. In Figure 6.26 the dynamic Part is segmented into two parts. More details can be found in [22].

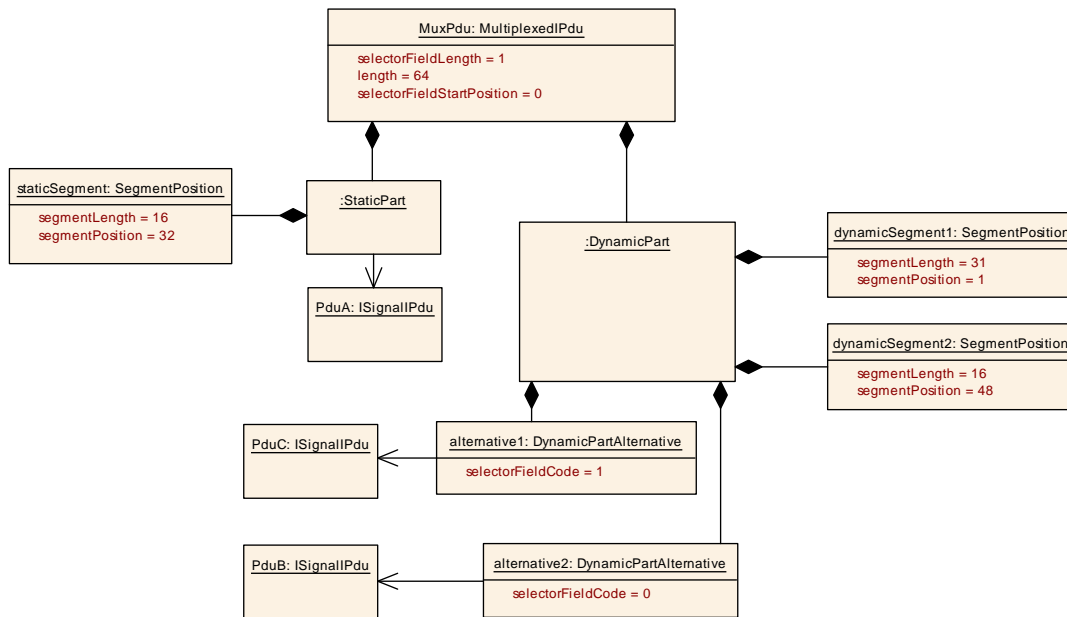


Figure 6.26: I-Pdu Multiplexer Example

Each of the following figures shows an example with an allowed IPduM configuration. Please note that the AUTOSAR IPduM module does not shift any part (static or dynamic) IPdu and just merges the payload. *ISignalIPdu*s that are referenced by the different *DynamicPartAlternatives* in one *MultiplexedIPdu* shall always have the same length. A configuration may be optimized with respect to unused data at end of a *StaticPart ISignalIPdu*. This is shown in figure 6.27 where the *ISignalIPdu* that is referenced by the *StaticPart* is shorter than the *MultiplexedIPdu*. An optimization with respect to unused data at end of *DynamicPartAlternative ISignalIPdu*s is shown in figure 6.28.

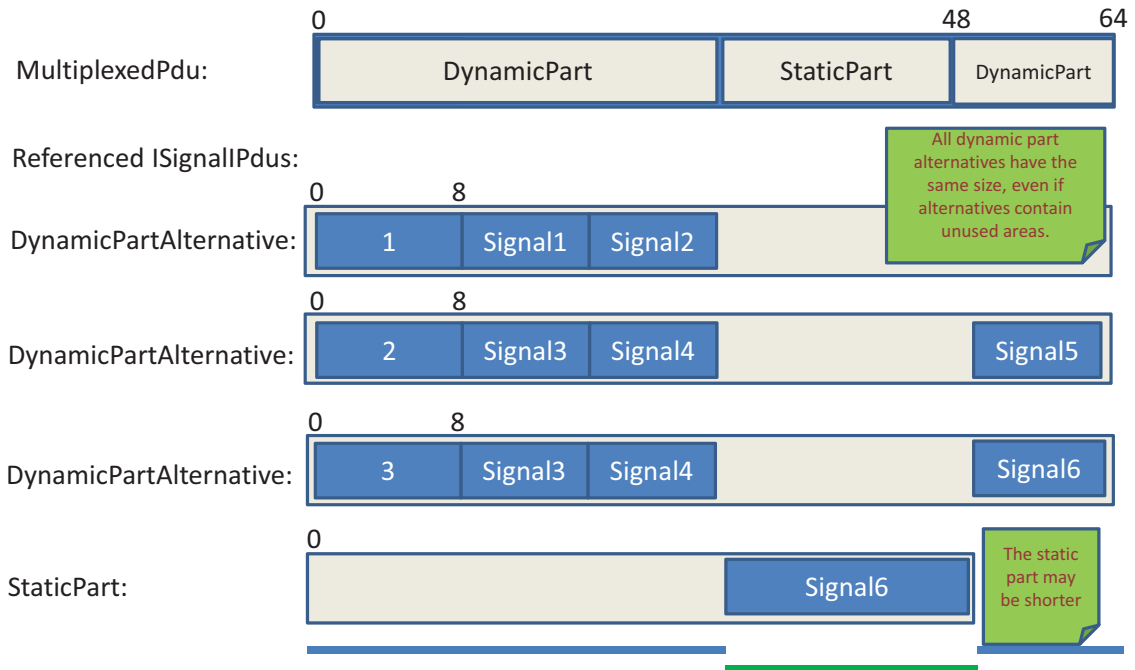


Figure 6.27: Multiplexer configuration example optimized with respect to unused data at end of static part Pdu

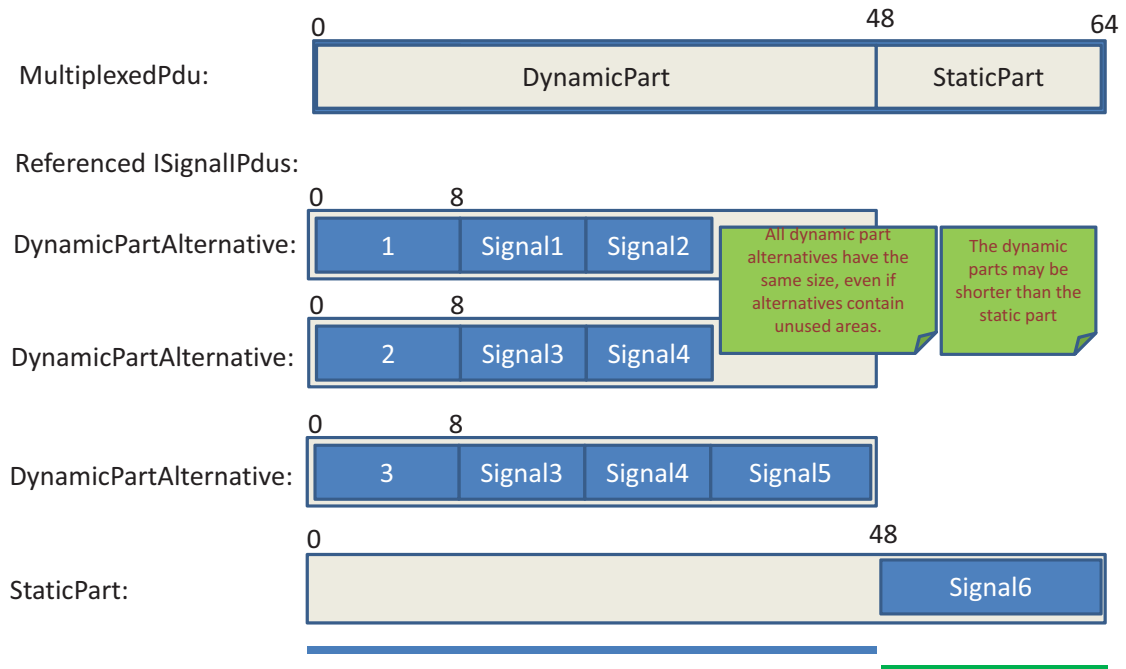


Figure 6.28: Multiplexer configuration example optimized with respect to unused data at end of dynamic part Pdu

6.5.1 I-Pdu Multiplexer in System Extract/ECU Extract

The processing in the ECU determines the description of `MultiplexedIPdu` in the System Extract/Ecu Extract. In case that a Gateway ECU only routes a `MultiplexedIPdu` without being interested in the content leads to a reduced description in the System Extract/ECU Extract. The following items describe the different scenarios and the consequences for the System Extract/ECU Extract description. A complete System Description contains all information.

[TPS_SYST_01080] Sending or receiving of a `MultiplexedIPdu` in System Extract/ECU Extract [

- all attributes of the `MultiplexedIPdu` are mandatory
- aggregated `DynamicPart` with associated `ISignalIPdu`s is mandatory in case
 - of sending
 - of receiving if at least one `DynamicPartAlternative` is received by one Ecu of the Extract.
- a `PduTriggering` shall be defined for the `MultiplexedIPdu`
- a `PduTriggering` shall be defined for all included `ISignalIPdu`s in the `DynamicPart` and `StaticPart`

]()

The initial ECU Configuration Generator configures COM, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01081] Gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract [

- `StaticPart` and `DynamicPart` definitions shall be omitted, thus no `ISignalIPdu` description shall be included
- all attributes of the `MultiplexedIPdu` shall be omitted.
- a `PduTriggering` shall be defined only for the gatewayed `MultiplexedIPdu`
- an `IPduMapping` between the source and the target `PduTriggerings` shall be defined

]()

The initial ECU Configuration Generator configures PduR and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01082] Receiving and gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract [

- all attributes of the `MultiplexedIPdu` are mandatory

- aggregated `DynamicPart` with associated `ISignalIPdus` is mandatory in case at least one `DynamicPartAlternative` is received by one Ecu of the Extract.
- a `PduTriggering` shall be defined for the `MultiplexedIPdu`
- an `IPduMapping` between the source and the target `PduTriggerings` shall be defined
- a `PduTriggering` shall be defined for all included `ISignalIPdus` in the `DynamicPart` and `StaticPart`

]()

The initial ECU Configuration Generator configures Com, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

6.6 Frames

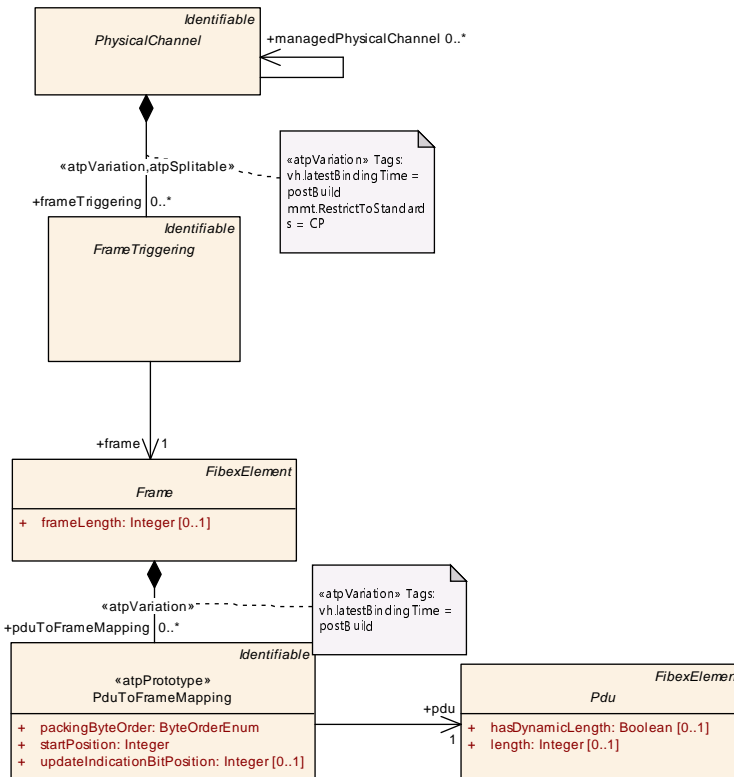


Figure 6.29: Frame Overview (FibexCore: FrameOverview)

[TPS_SYST_01083] **Frame** [A **Frame** represents a general design object that is used to describe the layout of the included **Pdus** as a reusable asset.]()

[TPS_SYST_01084] **FrameTriggering** [The **FrameTriggering** implements the reusable definition of a **Frame** within a concrete context and thus defines a **Frame**'s send behavior and identification on a certain **PhysicalChannel**.]()

[TPS_SYST_02255] **Frame.frameLength** usage for **FlexrayFrames** and **CanFrames** [The **frameLength** for a **FlexrayFrame** shall be equal or larger than the combined length of all **Pdus** that are mapped to the frame.

The **frameLength** for a **CanFrame** is used to describe the minimum length of a received L-PDU to be accepted by a data length check. Therefore, it is possible to configure a **frameLength** which is smaller than the mapped **Pdu** to this frame. If data length check is not needed the **frameLength** of a **CanFrame** may be left undefined. The reason for that is that if the **CanFrame.frameLength** is larger than the **Pdu.length** of the mapped **Pdu** and data length check is used, a received **Pdu** will always be discarded due to a failing minimum length check.]()

Class	Frame (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	<i>AbstractEthernetFrame</i> , <i>CanFrame</i> , <i>FlexrayFrame</i> , <i>LinFrame</i>			
Attribute	Type	Mult.	Kind	Note
frameLength	Integer	0..1	attr	The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay). The frameLength of zero bytes is allowed. Please consider also TPS_SYST_02255.
pduToFrame Mapping	PduToFrameMapping	*	aggr	A frames layout as a sequence of Pdus. atpVariation: The content of a frame can be variable. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.86: Frame

Class	FrameTriggering (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>CanFrameTriggering</i> , <i>EthernetFrameTriggering</i> , <i>FlexrayFrameTriggering</i> , <i>LinFrameTriggering</i>			
Attribute	Type	Mult.	Kind	Note
frame	Frame	1	ref	One frame can be triggered several times, e.g. on different channels. If a frame has no frame triggering, it won't be sent at all. A frame triggering has assigned exactly one frame, which it triggers.
framePort	FramePort	*	ref	References to the FramePort on every ECU of the system which sends and/or receives the frame. References for both the sender and the receiver side shall be included when the system is completely defined.
pduTriggering	PduTriggering	*	ref	This reference provides the relationship to the Pdu Triggerings that are implemented by the FrameTriggering. The reference is optional since no PduTriggering can be defined for NmPdus and XCP Pdus. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.87: FrameTriggering

6.7 Specialized Attributes of the Communication Entities

In the Basic Software the timing of bus frames can be controlled by send requests of the RTE in combination with the Transmission Mode and Transfer Property parameters in COM. On the other hand the timing can be controlled by the FlexRay Interface and LIN Interface.

This chapter describes the protocol specific extensions to the communication elements.

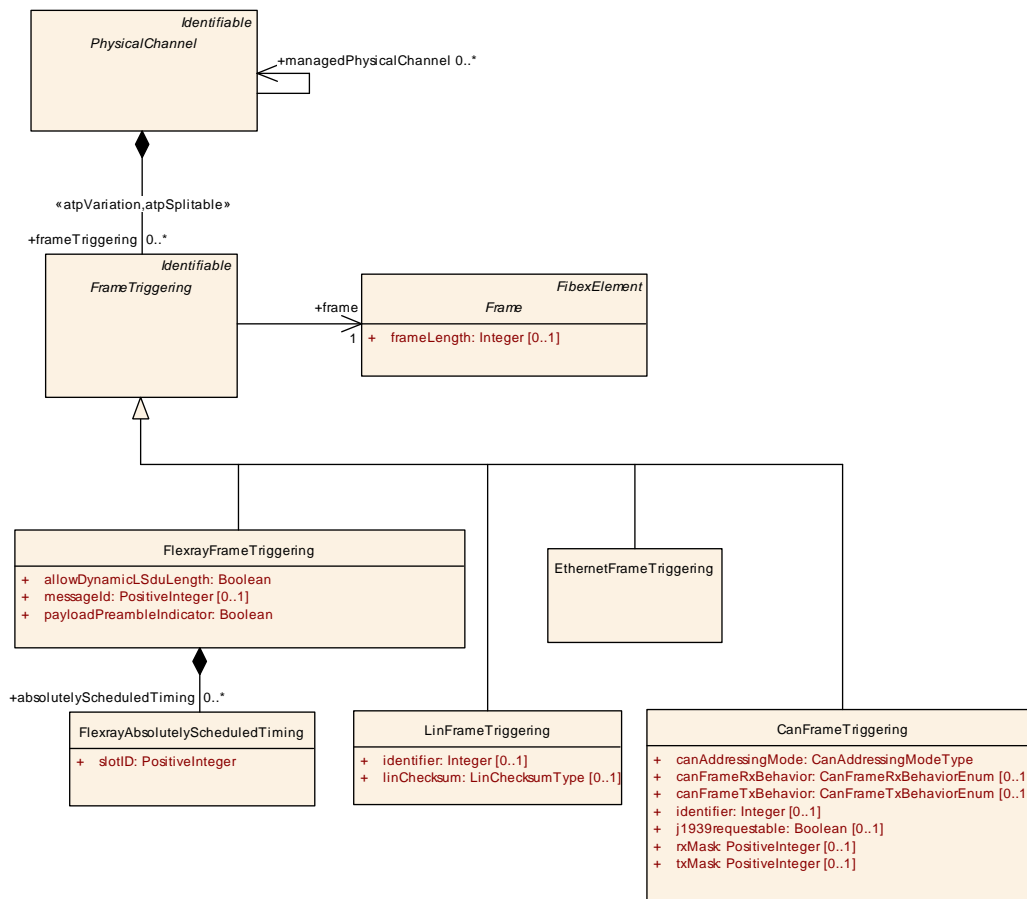


Figure 6.30: Frame Triggering

6.7.1 FlexRay specific description

[TPS_SYST_01128] Communication over FlexRay [The System Template supports the description of communication over FlexRay.] ([RS_SYST_00024](#))

In the following, the elements necessary to describe the FlexRay communication are specified.

FlexRay static segment parameters: Each `FlexrayFrameTriggering` is identified by its `slotID` and `communicationCycle`. In the static segment all communication

slots are of identical, statically configured duration and all `FrameTriggerings` are of identical, statically configured length.

The sending behavior where the exact time for the `FlexrayFrameTriggerings` transmission is guaranteed is provided in the System Template by the usage of `FlexrayAbsolutelyScheduledTiming`.

In the cycle counter field of every frame, the current value of the cycle counter is transmitted (see FlexRay frame format). This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.

[TPS_SYST_01085] Transmission of a `FrameTriggering` multiple times within one communication cycle [In the static segment `FlexrayFrameTriggerings` can be sent multiple times within one communication cycle. For describing this case multiple `FlexrayAbsolutelyScheduledTimings` shall be used.] (*RS_SYST_00024*)

FlexRay dynamic segment parameters: In the dynamic segment the duration of communication slots may vary in order to accommodate frames of varying length. Furthermore, in the dynamic part, the `slotID` is equivalent to a priority. The higher the number the lower is the priority.

The frames in the static and in the dynamic segment are described in the same way. Each `FlexrayFrameTriggering` is identified by its `slotID` and `communicationCycle`. A description is provided by the usage of `FlexrayAbsolutelyScheduledTiming`.

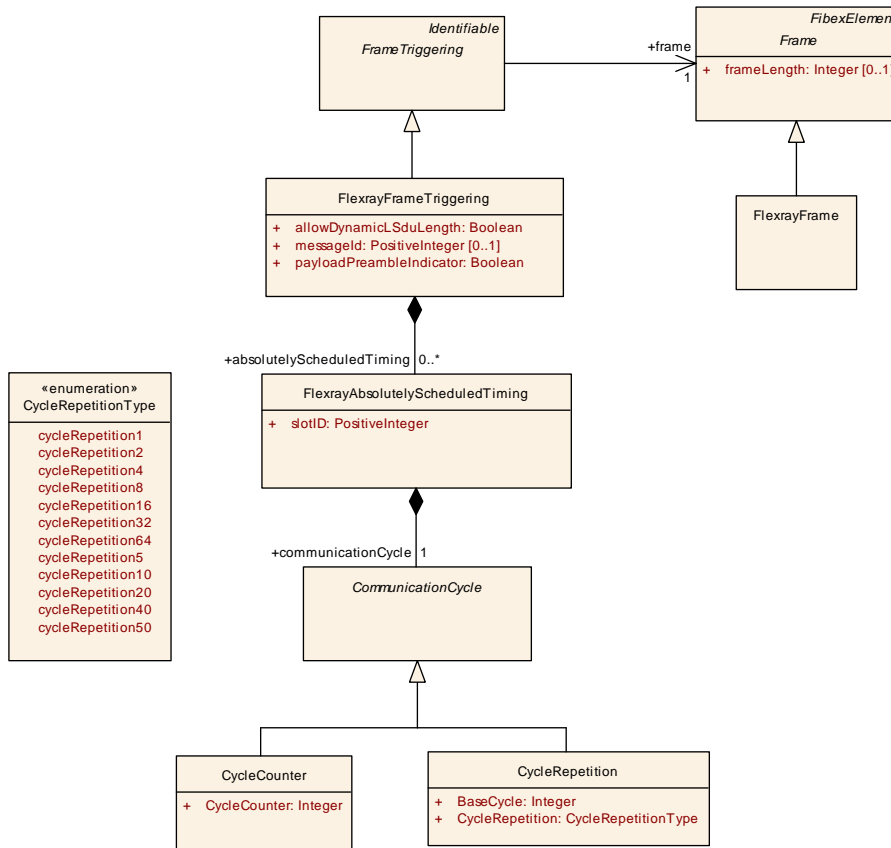


Figure 6.31: FlexRay Absolutely Scheduled Timing (Fibex4FlexRay:FlexrayAbsolutelyScheduledTiming)

Class	FlexrayFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication			
Note	FlexRay specific Frame element. Tags: atp.recommendedPackage=Frames			
Base	ARObject, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
-	-	-	-	-

Table 6.88: FlexrayFrame

Class	FlexrayFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication			
Note	FlexRay specific attributes to the FrameTriggering			
Base	ARObject, FrameTriggering, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
-	-	-	-	-





Class	FlexrayFrameTriggering			
absolutely Scheduled Timing	FlexrayAbsolutelyScheduledTiming	*	aggr	Specification of a sending behaviour where the exact time for the frames transmission is guaranteed.
allowDynamic LSduLength	Boolean	1	attr	Allows L-PDU length reduction and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU. If this attribute is set to true than the referenced Frame length attribute defines the max. length.
messageId	PositiveInteger	0..1	attr	The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.
payload Preamble Indicator	Boolean	1	attr	Switching the Payload Preamble bit.

Table 6.89: FlexrayFrameTriggering

Class	FlexrayAbsolutelyScheduledTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication			
Note	Each frame in FlexRay is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming. In the static segment a frame can be sent multiple times within one communication cycle. For describing this case multiple AbsolutelyScheduledTimings have to be used. The main use case would be that a frame is sent twice within one communication cycle.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
communication Cycle	CommunicationCycle	1	aggr	The communication cycle where the frame is sent.
slotID	PositiveInteger	1	attr	In the static part the SlotID defines the slot in which the frame is transmitted. The SlotID also determines, in combination with FlexrayCluster::numberOfStaticSlots, whether the frame is sent in static or dynamic segment. In the dynamic part, the slot id is equivalent to a priority. Lower dynamic slot ids are all sent until the end of the dynamic segment. Higher numbers, which were ignored that time, have to wait one cycle and then shall try again. minValue: 1 maxValue: 2047

Table 6.90: FlexrayAbsolutelyScheduledTiming

Class	CommunicationCycle (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is sent.			
Base	ARObject			
Subclasses	CycleCounter, CycleRepetition			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.91: CommunicationCycle

The communication cycle can be described by the [CycleCounter](#) or by the [CycleRepetition](#):

Class	CycleCounter			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is send is described by the attribute "cycleCounter".			
Base	ARObject, CommunicationCycle			
Attribute	Type	Mult.	Kind	Note
CycleCounter	Integer	1	attr	The communication cycle where the frame described by this timing is sent. If a timing is given in this way the referencing FlexrayCluster shall specify the cycleCount Max as upper bound and point of total repetition. This value is incremented at the beginning of each new cycle, ranging from 0 to cycleCountMax, and is reset to 0 after a sequence of cycleCountMax+1 cycles.

Table 6.92: CycleCounter

Class	CycleRepetition			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is send is described by the attributes baseCycle and cycle Repetition.			
Base	ARObject, CommunicationCycle			
Attribute	Type	Mult.	Kind	Note
BaseCycle	Integer	1	attr	The first communication cycle where the frame is sent. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.
CycleRepetition	CycleRepetitionType	1	attr	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.

Table 6.93: CycleRepetition

Enumeration	CycleRepetitionType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	The number of communication cycles (after the first cycle) whenever the frame is sent again. The FlexRay communication controller allows only determined values.
Literal	Description
cycleRepetition1	Attribute cycleRepetition value="1" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=0
cycleRepetition10	Attribute cycleRepetition value="10" to support FlexRay 3.0 Tags: atp.EnumerationLiteralIndex=1
cycleRepetition16	Attribute cycleRepetition value="16" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=2
cycleRepetition2	Attribute cycleRepetition value="2" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=3
cycleRepetition20	Attribute cycleRepetition value="20" to support FlexRay 3.0 Tags: atp.EnumerationLiteralIndex=4
cycleRepetition32	Attribute cycleRepetition value="32" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=5
cycleRepetition4	Attribute cycleRepetition value="4" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=6
cycleRepetition40	Attribute cycleRepetition value="40" to support FlexRay 3.0 Tags: atp.EnumerationLiteralIndex=7
cycleRepetition5	Attribute cycleRepetition value="5" to support FlexRay 3.0 Tags: atp.EnumerationLiteralIndex=8
cycleRepetition50	Attribute cycleRepetition value="50" to support FlexRay 3.0 Tags: atp.EnumerationLiteralIndex=9
cycleRepetition64	Attribute cycleRepetition value="64" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=10
cycleRepetition8	Attribute cycleRepetition value="8" valid only for FlexRay Protocol 2.1 Rev A Tags: atp.EnumerationLiteralIndex=11

Table 6.94: CycleRepetitionType

[constr_3012] Overlapping of Pdus is prohibited [Pdus mapped to a FlexrayFrame shall NOT overlap.]()

[constr_3013] FlexrayFrame length shall not be exceeded [The combined length of all Pdus that are mapped into a FlexrayFrame shall not exceed the defined FlexrayFrame length.]()

[constr_3014] Overlapping of updateIndicationBits for Pdus is prohibited [The updateIndicationBitPosition for a Pdu in a FlexrayFrame shall NOT overlap with other updateIndicationBitPositions and Pdu locations.]()

[constr_5104] Assignment of a FlexrayFrame where allowDynamicLSduLength is set to true [FlexrayFrames which are referenced by a FlexrayFrameTriggering where allowDynamicLSduLength is set to true shall always be assigned to the dynamic segment.]()

[constr_5105] Mapping of Pdu with dynamic length in a FlexrayFrame [Only the last Pdu in a FlexrayFrame is allowed to be a Pdu with hasDynamicLength = true.]()

Note: Please be aware that the dynamic Pdu at the end of the FlexRay Frame may need to provide some mechanism to determine its actual length (e.g. length field, termination). Otherwise the receiver is not able to determine the actual sent length.

6.7.2 LIN specific description

LIN is a protocol that is based on a single master - multiple slave principle. In the following, the parameters will be specified, which are necessary to describe the LIN Schedule Tables and the LIN Frames.

[TPS_SYST_01129] Communication over LIN [The System Template supports the description of communication over LIN.](*RS_SYST_00022*)

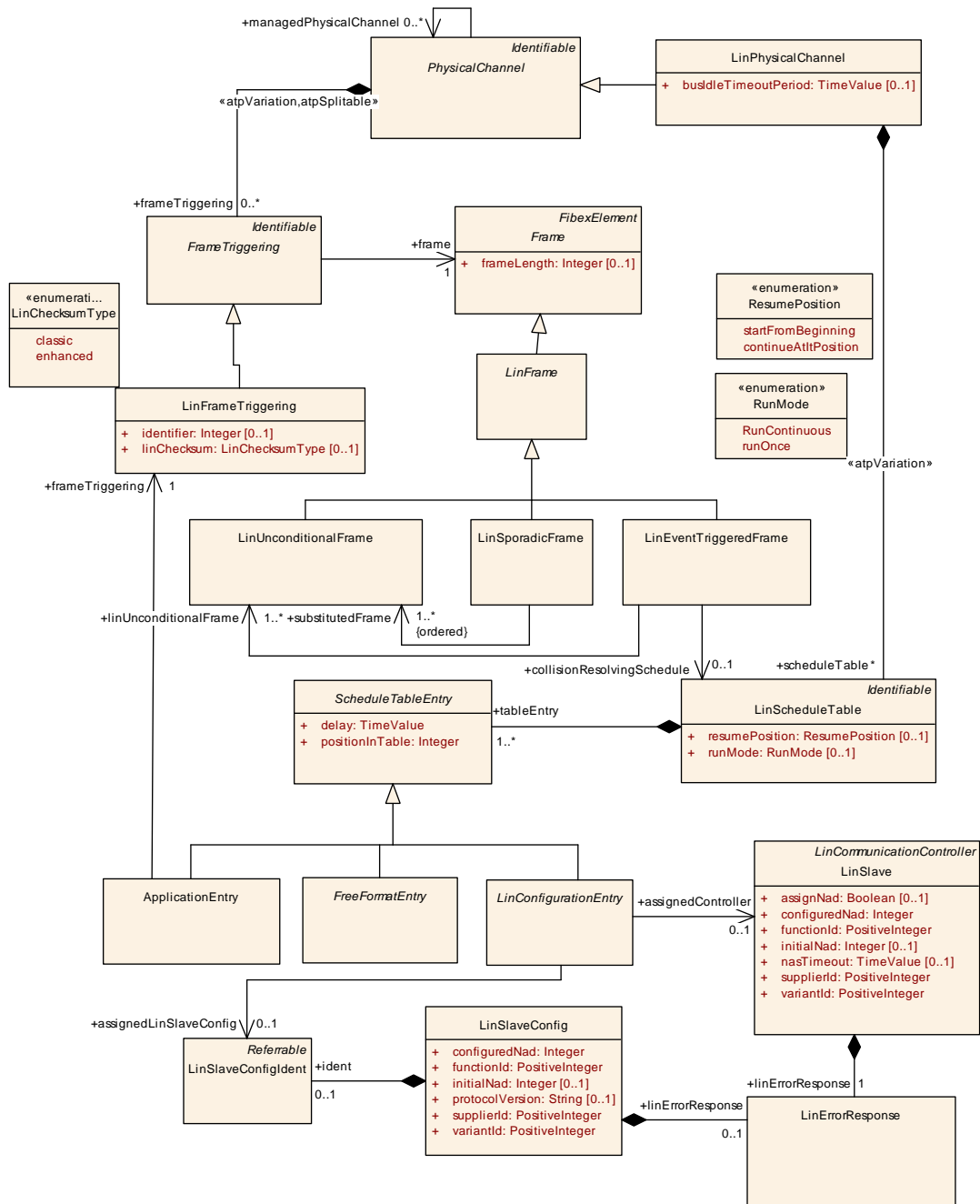


Figure 6.32: LIN Schedule Table (Fibex4Lin:LinScheduleTable)

6.7.2.1 LIN Frames

One LIN FrameMaster consists of two parts: header and response. The header is always sent by a **LinMaster**, while the response is sent by only one dedicated **LinSlave**. There are three different ways of transmitting frames on the bus: unconditional, event triggered, and sporadic frames.

Class	<i>LinFrame</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Lin specific Frame element.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Frame</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	<i>LinEventTriggeredFrame</i> , <i>LinSporadicFrame</i> , <i>LinUnconditionalFrame</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.95: LinFrame

Class	<i>LinFrameTriggering</i>			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	LIN specific attributes to the FrameTriggering			
Base	<i>ARObject</i> , <i>FrameTriggering</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
identifier	Integer	0..1	attr	To describe a frames identifier on the communication system, usually with a fixed identifierValue. For Lin SporadicFrames the attribute shall be ignored.
linChecksum	LinChecksumType	0..1	attr	Type of checksum that the frame is using. This attribute is optional because in case of sporadic frames it should not be set.

Table 6.96: LinFrameTriggering

Enumeration	<i>LinChecksumType</i>
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	Use of classic or enhanced checksum is managed by the master node and it is determined per frame identifier;
Literal	Description
classic	Classic in communication with LIN 1.3 slave nodes Tags: atp.EnumerationLiteralIndex=0
enhanced	Enhanced in communication with LIN 2.0 slave nodes. Tags: atp.EnumerationLiteralIndex=1

Table 6.97: LinChecksumType

[TPS_SYST_02095] [LinFrameTriggering.linChecksum](#) for [LinUnconditionalFrames](#) [The [linChecksum](#) attribute of a [LinFrameTriggering](#) that references a [LinUnconditionalFrame](#) shall be set.]()

Class	<i>LinUnconditionalFrame</i>
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data. Tags: atp.recommendedPackage=Frames





Class	LinUnconditionalFrame			
Base	ARObject, CollectableElement, FibexElement, Frame, Identifiable, LinFrame, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.98: LinUnconditionalFrame

[constr_3225] LinFrameTriggering.linChecksum not allowed for LinSporadicFrames [The linChecksum attribute of a LinFrameTriggering that references a LinSporadicFrame shall not be set.]()

[constr_3226] LinFrameTriggering.linChecksum for LinEventTriggeredFrames [Within a PhysicalChannel the linChecksum attribute of a LinFrameTriggering that references a LinEventTriggeredFrame shall have the same value as the linChecksum attribute of each LinFrameTriggering that references a LinUnconditionalFrame that in turn is referenced by that LinEventTriggeredFrame.]()

[constr_3203] LinFrameTriggering to LinSporadicFrame reference restriction in LinSporadicFrame context [Within a PhysicalChannel a LinUnconditionalFrame shall be referenced by only one LinFrameTriggering to allow a derivation of the identifier of a substituted Frame if the LinUnconditionalFrame is referenced by a LinSporadicFrame in the role substitutedFrame.]()

Class	LinSporadicFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pcus. Tags: atp.recommendedPackage=Frames			
Base	ARObject, CollectableElement, FibexElement, Frame, Identifiable, LinFrame, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
substituted Frame (ordered)	LinUnconditionalFrame	1..*	ref	Reference to a group of unconditional frames that share the same frame slot. In case that more than one of the declared frames needs to be transferred, the one first listed shall be chosen. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. A LinUnconditionalFrame associated with a LinSporadic Frame may not be allocated in the same LinSchedule Table as the sporadic frame.

Table 6.99: LinSporadicFrame

[constr_3204] LinUnconditionalFrames associated with a LinSporadicFrame [A LinUnconditionalFrame associated with a LinSporadicFrame

shall not be allocated in the same [LinScheduleTable](#) as the [LinSporadicFrame](#).
()

[constr_3205] Existence of [FramePort](#) for a [FrameTriggering](#) that references a [LinSporadicFrame](#) [A [FrameTriggering](#) that references a [LinSporadicFrame](#) shall not have a reference to a [FramePort](#).]()

Instead of the [LinSporadicFrame](#) a [LinUnconditionalFrame](#) is sent in the timeslot on the bus and therefore the [FrameTriggering](#) that references a [LinSporadicFrame](#) does not need to have a reference to a [FramePort](#).

Class	LinEventTriggeredFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	<p>An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response.</p> <p>The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable.</p> <p>The event controlled frame shall not contain any Pdus.</p> <p>Tags:atp.recommendedPackage=Frames</p>			
Base	<i>ARObject, CollectableElement, FibexElement, Frame, Identifiable, LinFrame, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
collisionResolvingSchedule	LinScheduleTable	0..1	ref	Reference to the schedule table, which resolves a collision.
linUnconditionalFrame	LinUnconditionalFrame	1..*	ref	<p>A list of slaves can respond to the master request if at least one of the signals carried in its unconditional frame is updated.</p> <p>For each response a LinFrameTriggering and a LinUnconditionalFrame shall be defined. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element.</p> <p>The Unconditional frames associated with an event triggered frame shall:</p> <ul style="list-style-type: none"> • have equal length. • use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed). • reserve the first data field to its protected identifier (even if the associated unconditional frame is scheduled as a unconditional frame in the same or another schedule table). • be published by different slave nodes. • shall not be included directly in the same schedule table as the event triggered frame is scheduled.

Table 6.100: LinEventTriggeredFrame

[TPS_SYST_02077] Subscribers of a [LinEventTriggeredFrame](#) [For each subscriber of a [LinEventTriggeredFrame](#) a [LinUnconditionalFrame](#) and a [LinFrameTriggering](#) that points to this [LinUnconditionalFrame](#) shall be defined.]()

[constr_3202] LinFrameTriggering to LinUnconditionalFrame reference restriction in LinEventTriggeredFrame context [Within a `PhysicalChannel` a `LinUnconditionalFrame` shall be referenced by only one `LinFrameTriggering` to allow a derivation of the identifier of a substituted Frame if the `LinUnconditionalFrame` is referenced by a `LinEventTriggeredFrame` in the role `linUnconditionalFrame`.]()

[constr_3206] Existence of FramePort for a FrameTriggering that references a LinEventTriggeredFrame [A `FrameTriggering` that references a `LinEventTriggeredFrame` shall not have a reference to a `FramePort`.]()

A `LinUnconditionalFrame` is sent as the response of a `LinEventTriggeredFrame` on the bus instead and therefore the `FrameTriggering` that references a `LinEventTriggeredFrame` does not need to have a reference to a `FramePort`.

[TPS_SYST_02078] LinUnconditionalFrames associated with a LinEventTriggeredFrame [The `LinUnconditionalFrames` associated with a `LinEventTriggeredFrame` shall:

- have equal length
- use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed)
- reserve the first data field to its protected identifier (even if the associated `LinUnconditionalFrame` is scheduled as a `LinUnconditionalFrame` in the same or another schedule table)
- be published by different slave nodes
- not be included directly in the same `LinScheduleTable` as the associated `LinEventTriggeredFrame`.

]()

6.7.2.2 LIN Schedule Table

The `LinMaster` uses one or more predefined scheduling tables to start the sending and receiving to the LIN bus. These scheduling tables contain at least the relative timing that defines the message sending.

[constr_1657] Existence of LinPhysicalChannel.scheduleTable [In any given Ecu Extract that contains a `LinSlave`, the `LinPhysicalChannel` that relates to the respective `LinSlave` via `commConnector.commController` shall not aggregate a `LinScheduleTable`.]()

Class	LinScheduleTable			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
resumePosition	ResumePosition	0..1	attr	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
runMode	RunMode	0..1	attr	The schedule table can be executed in two different modes.
tableEntry	ScheduleTableEntry	1..*	aggr	The scheduling table consists of table entries, which contain Frame slots.

Table 6.101: LinScheduleTable

Enumeration	RunMode
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	The schedule table can be executed in two different modes.
Literal	Description
RunContinuous	RUN_CONTINUOUS run mode Tags: atp.EnumerationLiteralIndex=0
runOnce	RUN_ONCE run mode Tags: atp.EnumerationLiteralIndex=1

Table 6.102: RunMode

Enumeration	ResumePosition
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
Literal	Description
continueAtItPosition	Continue at IT Point. Tags: atp.EnumerationLiteralIndex=0
startFromBeginning	Start from the beginning Tags: atp.EnumerationLiteralIndex=1

Table 6.103: ResumePosition

Class	ScheduleTableEntry (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Table entry in a LinScheduleTable. Specifies what will be done in the frame slot.			
Base	ARObject			
Subclasses	ApplicationEntry , FreeFormatEntry , LinConfigurationEntry			
Attribute	Type	Mult.	Kind	Note
delay	TimeValue	1	attr	Relative delay between this tableEntry and the start of the successor in the schedule table in seconds.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the schedule table entry. Tags: xml.sequenceOffset=-10
positionInTable	Integer	1	attr	Relative position in the schedule table. The first entry index in the schedule table is 0.

Table 6.104: ScheduleTableEntry

Class	ApplicationEntry			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule table entry for application messages.			
Base	ARObject, ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
frameTriggering	LinFrameTriggering	1	ref	Specifies the LinFrame that will be transmitted in this frame slot.

Table 6.105: ApplicationEntry

Class	FreeFormatEntry (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	FreeFormat transmits a fixed master request frame with the eight data bytes provided. This may for instance be used to issue user specific fixed frames.			
Base	ARObject, ScheduleTableEntry			
Subclasses	FreeFormat			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.106: FreeFormatEntry

Class	LinConfigurationEntry (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	A ScheduleTableEntry which contains LIN specific assignments.			
Base	ARObject, ScheduleTableEntry			
Subclasses	AssignFrameId , AssignFrameIdRange , AssignNad , ConditionalChangeNad , DataDumpEntry , SaveConfigurationEntry , UnassignFrameId			
Attribute	Type	Mult.	Kind	Note





Class	<i>LinConfigurationEntry</i> (abstract)			
assigned Controller	LinSlave	0..1	ref	The LIN slaves controller who is target of this assignment. Optional in case LinConfigurationEntry.assignedLinSlave Config exists.
assignedLin SlaveConfig	LinSlaveConfigIdent	0..1	ref	The LIN slave that is target of this assignment. Please note that this reference is redundant to the assignedController reference. In an Ecu Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the LinMaster.

Table 6.107: LinConfigurationEntry

6.7.2.3 Configuration Services

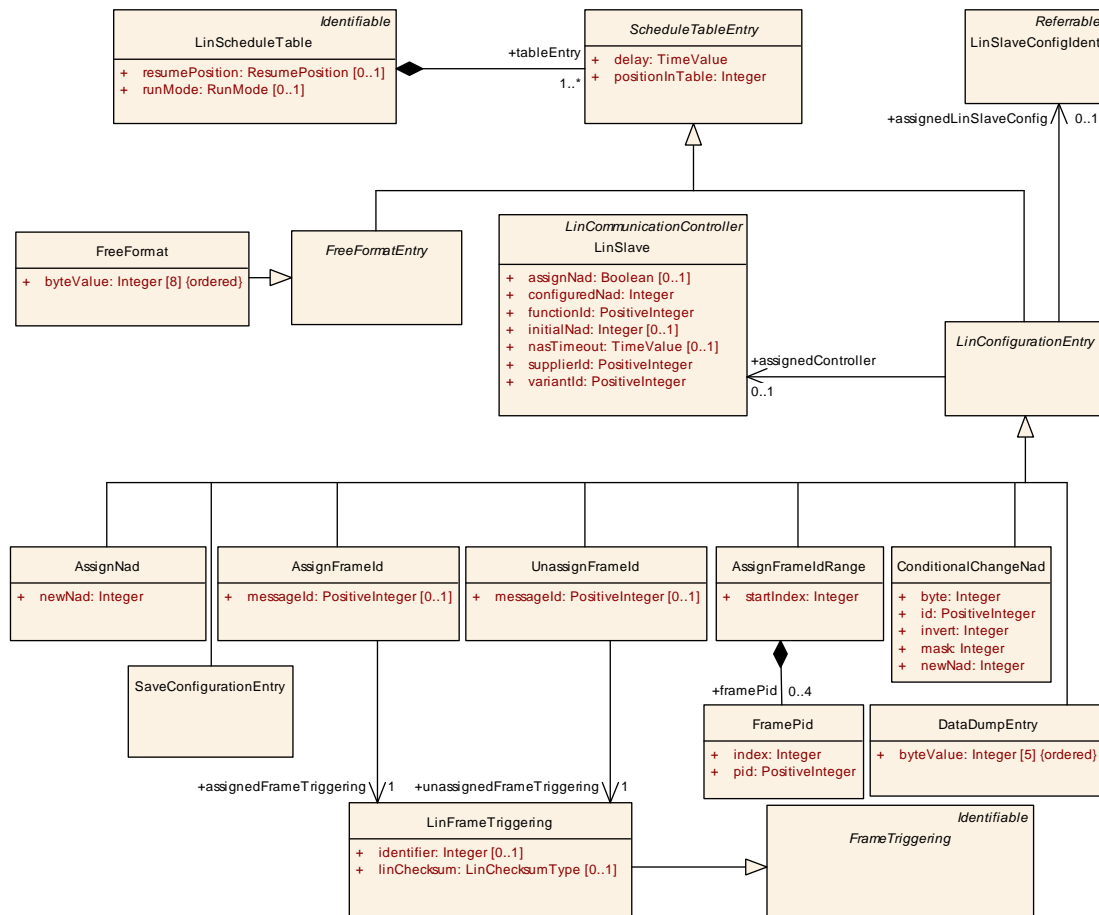


Figure 6.33: LIN Configuration Entries (Fibex4Lin:LinConfigurationEntries)

LIN only supports 64 identifiers. That creates the need for extending the address space. Hence the frames are identified by message ids from a much larger address

space that is additionally separated by supplier ids. During runtime the master assigns a LinId to the frame. In case of identical parts within a cluster the initial node ID (oldNad) is used to differentiate such nodes.

To support that in System Template the [AssignFrameId](#) is introduced as a LIN specific extension. For the assignment a relation to the [LinSlave](#) is used. The [LinSlave](#) element is referenced by a [LinCommunicationConnector](#) element that contains a list of frames processed by the slave node. More details can be found in chapter [6.7.2.3](#).

Class	AssignFrameId			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Assign Frame Id master request.			
Base	ARObject , LinConfigurationEntry , ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
assignedFrameTriggering	LinFrameTriggering	1	ref	The frame whose identifier is set by this assignment.
messageId	PositiveInteger	0..1	attr	MessageId of the referenced frame.

Table 6.108: AssignFrameId

Class	UnassignFrameId			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier.			
Base	ARObject , LinConfigurationEntry , ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
messageId	PositiveInteger	0..1	attr	MessageId of the referenced frame.
unassignedFrameTriggering	LinFrameTriggering	1	ref	The frame whose identifier is reset by this assignment.

Table 6.109: UnassignFrameId

[constr_3241] Usage of [AssignFrameId.messageId](#) [The value of [AssignFrameId.messageId](#) for the [AssignFrameId](#) that refers to a [LinSlave](#) in the role [assignedController](#) shall be equal to the [messageId](#) of the [LinConfigurableFrame](#) aggregated by [LinCommunicationConnector](#) in role [linConfigurableFrame](#) that points to this [LinSlave](#) in the role [commController](#).]()

[constr_3242] Usage of [UnassignFrameId.messageId](#) [The value of [UnassignFrameId.messageId](#) for the [UnassignFrameId](#) that refers to a [LinSlave](#) in the role [assignedController](#) shall be equal to the [messageId](#) of the [LinConfigurableFrame](#) aggregated by [LinCommunicationConnector](#) in role [linConfigurableFrame](#) that points to this [LinSlave](#) in the role [commController](#).]()

The Assign frame ID configuration service is replaced in LIN 2.1 by the Assign frame ID range configuration service. [AssignFrameIdRange](#) is used to set or disable Protected Identifiers up to four frames. For the assignment a relation to the [LinSlave](#) is used. The [LinSlave](#) element is referenced by a [LinCommunicationConnector](#)

element that contains a list of frames processed by the slave node. More details can be found in chapter 6.7.2.3.

Class	AssignFrameIdRange			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	AssignFrameIdRange generates an assign frame PID range request.			
Base	<i>ARObject</i> , <i>LinConfigurationEntry</i> , <i>ScheduleTableEntry</i>			
Attribute	Type	Mult.	Kind	Note
framePid	FramePid	0..4	aggr	Optional assignment of frame_PID values that are included in the request. The frame_PIDs are ordered.
startIndex	Integer	1	attr	The startIndex sets the index to the first frame to assign a PID.

Table 6.110: AssignFrameIdRange

Class	FramePid			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Frame_PIDs that are included in the request. The "pid" attribute describes the value and the "index" attribute the position of the frame_PID in the request.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
index	Integer	1	attr	This attribute is used to order the frame_PIDs. The values of index shall be unique within one AssignFrameIdRange.
pid	PositiveInteger	1	attr	Frame_PID value.

Table 6.111: FramePid

[constr_5031] Uniqueness of [FramePid.index](#) [[FramePid.index](#) shall always be set and be unique in the context of the aggregating [AssignFrameIdRange](#).]()

Assign NAD is used to resolve conflicting NADs in LIN clusters built using off-the-shelves slave nodes or reused slave nodes. This request uses the initial NAD. The NAD used for the response shall be the same as in the request, i.e. the initial NAD.

Class	AssignNad			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Assign NAD master request.			
Base	<i>ARObject</i> , <i>LinConfigurationEntry</i> , <i>ScheduleTableEntry</i>			
Attribute	Type	Mult.	Kind	Note
newNad	Integer	1	attr	The newly assigned NAD value.

Table 6.112: AssignNad

The conditional change NAD is used to detect unknown slave nodes in a cluster and to separate their NADs.

Class	ConditionalChangeNad			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Generates an conditional change NAD request. See ISO 17987 protocol specification for more information.			
Base	ARObject , LinConfigurationEntry , ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
byte	Integer	1	attr	Byte Position of Data Byte that should be used for the bitwise XOR with Invert and the bitwise AND with Mask.
id	PositiveInteger	1	attr	Byte Position of Id.
invert	Integer	1	attr	Byte Position of Invert.
mask	Integer	1	attr	Byte Position of Mask.
newNad	Integer	1	attr	The newly assigned NAD value (Byte Position).

Table 6.113: ConditionalChangeNad

The Save Configuration service tells the slave node that the slave application shall save the current configuration.

Class	SaveConfigurationEntry			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	This service is used to notify a slave node to store its configuration.			
Base	ARObject , LinConfigurationEntry , ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.114: SaveConfigurationEntry

The Data Dump service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific.

Class	DataDumpEntry			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	This service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific.			
Base	ARObject , LinConfigurationEntry , ScheduleTableEntry			
Attribute	Type	Mult.	Kind	Note
byteValue (ordered)	Integer	5	attr	Supplier specific format.

Table 6.115: DataDumpEntry

With the FreeFormat a scheduling of fixed data content within a diagnostic frame is defined. For that specification [FreeFormat](#) is introduced.

Class	FreeFormat			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Representing freely defined data.			
Base	<i>ARObject</i> , <i>FreeFormatEntry</i> , <i>ScheduleTableEntry</i>			
Attribute	Type	Mult.	Kind	Note
byteValue (ordered)	Integer	8	attr	The integer Value of a freely defined data byte.

Table 6.116: FreeFormat

In order to be consistent with the rest of the communication configuration, it is required that the diagnostic LIN Frames (Master Request Frame, Slave Response Frame) are explicitly modeled as *Frame* elements. *LinFrameTriggerings* dealing with diagnostic Frames thus reference this diagnostic frames.

6.7.3 CAN specific description

This chapter describes additions to the CAN definition of `FrameTriggerings`.

[TPS_SYST_01130] **Communication over CAN** [The System Template supports the description of communication over CAN.] (*RS_SYST_00021*)

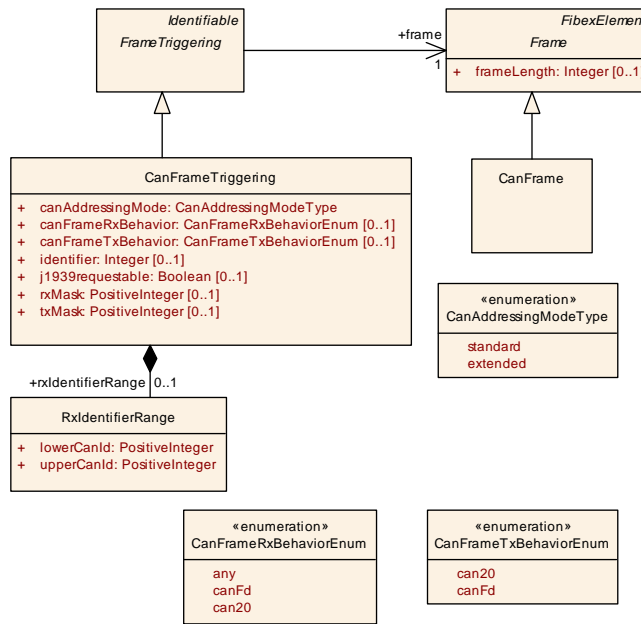


Figure 6.34: CanFrameTriggering (Fibex4Can:CanCommunication)

Class	CanFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific Frame element. This element shall also be used for TTCan. Tags: atp.recommendedPackage=Frames			
Base	ARObject, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.117: CanFrame

Class	CanFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific attributes to the FrameTriggering			
Base	<i>ARObject</i> , <i>FrameTriggering</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
absolutely Scheduled Timing	TtcanAbsolutelyScheduledTiming	*	aggr	Each frame in TTCAN is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming.
canAddressing Mode	CanAddressingModeType	1	attr	The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.
canFrameRx Behavior	CanFrameRxBehaviorEnum	0..1	attr	Defines which CAN protocol shall be expected for frame reception.
canFrameTx Behavior	CanFrameTxBehaviorEnum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
identifier	Integer	0..1	attr	This attribute is used to define the identifier this frame shall use on the CAN network.
j1939requestable	Boolean	0..1	attr	Frame can be triggered by the J1939 request message.
rxIdentifier Range	RxIdentifierRange	0..1	aggr	Optional definition of a CanId range.
rxMask	PositiveInteger	0..1	attr	Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range.
txMask	PositiveInteger	0..1	attr	Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.

Table 6.118: CanFrameTriggering

Enumeration	CanAddressingModeType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Indicates whether standard or extended CAN identifiers are used
Literal	Description
extended	Extended 29-bit-identifiers are used (CAN 2.0B) Tags: atp.EnumerationLiteralIndex=0
standard	Standard 11-bit-identifiers are used (CAN 2.0A) Tags: atp.EnumerationLiteralIndex=1

Table 6.119: CanAddressingModeType

Class	RxIdentifierRange
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Optional definition of a CanId range to reduce the effort of specifying every possible FrameTriggering within the defined Id range during reception. All frames received within a range are mapped to the same Pdu that is passed to a upper layer module (e.g. Nm, CDD, PduR).
Base	<i>ARObject</i>





Class		RxIdentifierRange		
Attribute	Type	Mult.	Kind	Note
lowerCanId	PositiveInteger	1	attr	This attribute can be used together with the upperCanId attribute to define a range of CanIds.
upperCanId	PositiveInteger	1	attr	This attribute can be used together with the lowerCanId attribute to define a range of CanIds.

Table 6.120: RxIdentifierRange

Enumeration	CanFrameRxBehaviorEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Defines different CAN protocols for frame reception behavior.
Literal	Description
any	This CAN frame may be received as both, CAN 2.0 and CAN FD. Tags: atp.EnumerationLiteralIndex=0
can20	This CAN frame shall be received as CAN 2.0 only. In case the CAN frame is received as CAN FD it is discarded during reception. Tags: atp.EnumerationLiteralIndex=1
canFd	This CAN frame shall be received as CAN FD only. In case the CAN frame is received as CAN 2.0 it is discarded during reception. Tags: atp.EnumerationLiteralIndex=2

Table 6.121: CanFrameRxBehaviorEnum

There exist use-cases where the `CanFrameTriggering` is used as a placeholder for a variant number of actual Can frames and therefore no dedicated CAN identifier can be defined (e.g. MetaData handling, Bus Mirroring).

[TPS_SYST_02201] Existence of `CanFrameTriggering.identifier` [In a `System` with `category` `SYSTEM_DESCRIPTION` the identifier may be omitted if the value is computed during runtime. In a `System` with `category` `ECU_EXTRACT`, `ECU_SYSTEM_DESCRIPTION` or `SYSTEM_EXTRACT` for the transmitter the `identifier` may be omitted if the value is computed during runtime. In an `System` with `category` `ECU_EXTRACT`, `ECU_SYSTEM_DESCRIPTION` or `SYSTEM_EXTRACT` for the receiver the `identifier` may be omitted if `rxIdentifierRange` is defined.]()

[TPS_SYST_02256] Allowed `CanFrame.frameLength` settings [For a `CanFrame` it is allowed to configure a smaller `frameLength` than the `length` of the `Pdu` which is mapped to this `CanFrame`. This is used to model the minimum length of the received L-PDU of a `CanFrame` to be accepted, if the data length check is enabled.]()

The `CanFrameTriggering.canFrameRxBehavior` allows to define a tolerant CAN FD reception strategy. With the setting `any` the respective CAN frame is accepted for reception, regardless whether it is received with CAN FD or CAN 2.0 protocol.

Enumeration	CanFrameTxBehaviorEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Defines different CAN protocols for frame transmission behavior.
Literal	Description
can20	This CAN frame shall be sent as CAN 2.0 only. Tags: atp.EnumerationLiteralIndex=0
canFd	This CAN frame shall be sent as CAN FD. Tags: atp.EnumerationLiteralIndex=1

Table 6.122: CanFrameTxBehaviorEnum

Note that the transmission behavior of [CanFrameTriggering.canFrameTxBehavior](#) may still be redefined in the communication stack on driver level.

[TPS_SYST_02168] MetaData support required if [CanFrameTriggering.txMask](#) is used [The usage of [CanFrameTriggering.txMask](#) requires the support of COM Stack MetaData.]()

Please note that the MetaData support in [\[TPS_SYST_02168\]](#) is required to calculate CAN-Ids at run-time.

[TPS_SYST_02169] MetaData support may be required if [CanFrameTriggering.rxMask](#) is used [The usage of [CanFrameTriggering.rxMask](#) may require the support of COM Stack MetaData.]()

Please note that the MetaData support in [\[TPS_SYST_02169\]](#) is required if the upper layer is interested in the masked part of CAN-Id, e.g. J1939. In some cases the upper layer is not interested in the masked part of CAN-Id, e.g. for CanNm the MetaData is not required.

6.7.3.1 SAE J1939 Protocol specific description

J1939 is a protocol and application layer standard of the SAE (Society of Automotive Engineers) based on the CAN technology. It defines parameters uniquely identified by the SPN (Suspect Parameter Number). These are mapped to parameter groups that are uniquely identified by a PGN (Parameter Group Number). Parameters are simply handled as `SystemSignals` which have a name derived from the name of the SPNs. A Parameter Group (PG) corresponds to an `IPdu`.

J1939 uses extended 29 bit CAN identifiers to encode a priority, the source address of the frame, and a frame ID which is based on the PGN (Parameter Group Number) and may contain the destination address.

J1939 supports `IPdus` with more than 8 bytes, and `IPdus` with variable length that may exceed 8 bytes. As soon as an `IPdu` has more than 8 bytes, it does not fit in a single CAN frame and a transport protocol shall be used. Variable length `IPdus` will always be handled by the J1939 TP, regardless of the actual length. The J1939 Transport Protocol is described in chapter 6.8.8.

[TPS_SYST_01132] Communication over SAE J1939 [The System Template supports the description of communication over SAE J1939.] ([RS_SYST_00038](#))

[constr_3209] CanFrameTriggerings with identical PGN [For all `CanFrameTriggerings` where the attribute `identifier` contains the identical PGN (as defined in section 5.2 Protocol Data Unit in [23]) the attribute `j1939requestable` shall also have an identical value.]()

6.7.4 TTCAN specific description

This chapter describes additions to the TTCAN definition of `FrameTriggerings`.

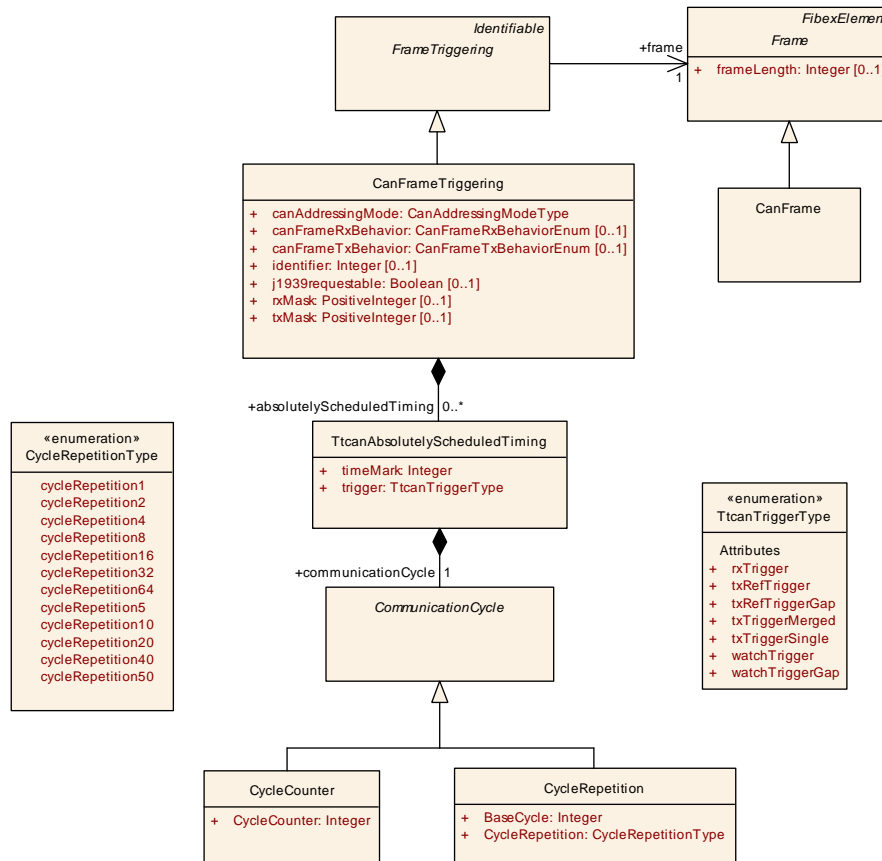


Figure 6.35: TtcanAbsolutelyScheduledTiming (Fibex4Ttcan:TtcanCommunication)

Class	TtcanAbsolutelyScheduledTiming			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanCommunication			
Note	<p>Each frame in TTCAN is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming.</p> <p>A frame can be sent multiple times within one communication cycle. For describing this case multiple AbsolutelyScheduledTimings have to be used. The main use case would be that a frame is sent twice within one communication cycle.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
communication Cycle	CommunicationCycle	1	aggr	The communication cycle where the frame is sent.
timeMark	Integer	1	attr	Where FlexRay counts the slots in the static segment, TTCAN requires explicit Tx and Rx time marks.
trigger	TtcanTriggerType	1	attr	Trigger type for this time window.

Table 6.123: TtcanAbsolutelyScheduledTiming

Enumeration	TtcanTriggerType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanCommunication
Note	This type lists all trigger types for a time window.
Literal	Description
rxTrigger	<p>Check for message reception</p> <p>Tags:atp.EnumerationLiteralIndex=0</p>
txRefTrigger	<p>Send reference message in periodic case</p> <p>Tags:atp.EnumerationLiteralIndex=1</p>
txRefTriggerGap	<p>Send reference message in event-synchronised case</p> <p>Tags:atp.EnumerationLiteralIndex=2</p>
txTriggerMerged	<p>Send message in a merged arbitration window</p> <p>Tags:atp.EnumerationLiteralIndex=3</p>
txTriggerSingle	<p>Send message in an exclusive time window</p> <p>Tags:atp.EnumerationLiteralIndex=4</p>
watchTrigger	<p>Check for missing reference message in periodic case</p> <p>Tags:atp.EnumerationLiteralIndex=5</p>
watchTriggerGap	<p>Check for missing reference message in event-synchronised case</p> <p>Tags:atp.EnumerationLiteralIndex=6</p>

Table 6.124: TtcanTriggerType

6.7.5 Ethernet specific description

Important note: Please note that the model of Release 4.4.0 to describe the Ethernet communication can still be used in Release 19-11. Elements like `SocketConnectionBundle` and `SocketConnection` are set to obsolete and will be removed in a future release. The documentation of the old model is available in the Rel. 4.4.0 System Template specification. For the usage of the Rel. 4.4.0 model only the attributes that are described in Release 4.4.0 are valid. A mixture of the Rel. 19-11 and Rel. 4.4.0 models is not allowed.

This chapter specifies how the data communication between nodes in an IP network over TCP and UDP protocols is described with an AUTOSAR model.

[TPS_SYST_01091] Definition of `SoAdConfig` [The `SoAdConfig` in the System Template is defined per `EthernetPhysicalChannel` which represents a VLAN.] ([RS_SYST_00039](#))

The `SoAdConfig` element is the entry point for the description of the IP communication on a VLAN since it contains a collection of `SocketAddresses` of nodes that are able to receive and transmit information over the VLAN. Each node is represented by an `EcuInstance`. The `SocketAddress` defines a communication endpoint (IP Unicast or IP Multicast) and assigns it to one `EthernetCommunicationConnector` of an `EcuInstance` in case of IP Unicast communication and to one or several `EthernetCommunicationConnectors` in case of IP Multicast.

Class	SoAdConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	SoAd Configuration for one specific Physical Channel.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
connection	SocketConnection	*	aggr	This aggregation is obsolete and will be removed in the future. The <code>connectionGroup</code> aggregation with bundled Connections shall be used instead. Old description: Collection of socket connections. Stereotypes: <code>atpVariation</code> Tags: <code>atp.Status=obsolete</code> <code>vh.latestBindingTime=postBuild</code>
connection Bundle	SocketConnection Bundle	*	aggr	Collection of <code>SocketConnectionBundles</code> . Stereotypes: <code>atpVariation</code> Tags: <code>atp.Status=obsolete</code> <code>vh.latestBindingTime=postBuild</code>
socketAddress	SocketAddress	1..*	aggr	Collection of <code>SoAdAddresses</code> . Stereotypes: <code>atpVariation</code> Tags: <code>vh.latestBindingTime=postBuild</code>

Table 6.125: SoAdConfig

Class		SocketAddress		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This meta-class represents a socket address towards the rest of the meta-model. The actual semantics of the represented socket address, however, is contributed by aggregation of an <code>ApplicationEndpoint</code> .			
Base	<code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
allowedIPv6Ext Headers	<code>IPv6ExtHeaderFilterList</code>	0..1	ref	Reference to a list of IPv6 Extension Headers allowed for this <code>SocketConnection</code> . If no list is referenced all IPv6 Extension Headers are allowed and processed.
allowedTcp Options	<code>TcpOptionFilterList</code>	0..1	ref	Reference to a list of TCP options allowed for this <code>SocketConnection</code> .
application Endpoint	<code>ApplicationEndpoint</code>	0..1	aggr	Application addressing
connector	<code>EthernetCommunicationConnector</code>	0..1	ref	Association to a <code>CommunicationConnector</code> in the topology description. This reference shall be used if the <code>SocketAddress</code> describes an IP unicast address for an ECU that is part of the model.
differentiated ServiceField	<code>PositiveInteger</code>	0..1	attr	The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.
flowLabel	<code>PositiveInteger</code>	0..1	attr	The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.
multicast Connector	<code>EthernetCommunicationConnector</code>	*	ref	Association to a <code>CommunicationConnector</code> in the topology description. This reference shall be used if the <code>SocketAddress</code> describes an IP multicast address. This multicast <code>SocketAddress</code> contains references to those ECUs in the model that want to receive the multicast messages.
pathMtu Discovery Enabled	<code>Boolean</code>	0..1	attr	Defines whether the Path MTU Discovery shall be performed for the related socket.
pduCollection MaxBufferSize	<code>PositiveInteger</code>	0..1	attr	Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.
pduCollection Timeout	<code>TimeValue</code>	0..1	attr	Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.
staticSocket Connection	<code>StaticSocketConnection</code>	*	aggr	Definition of a static <code>SocketConnection</code> . Stereotypes: <code>atpVariation</code> Tags: <code>vh.latestBindingTime=postBuild</code>
udpChecksum Handling	<code>UdpChecksumCalculationEnum</code>	0..1	attr	Specifies if UDP checksum handling shall be enabled (<code>udpChecksumEnabled</code>) or skipped (<code>udpChecksumDisabled</code>) on the related socket connection.

Table 6.126: SocketAddress

The communication endpoint itself is defined by the `ApplicationEndpoint` that is aggregated by the `SocketAddress` and by the `NetworkEndpoint` that in turn is referenced by the `ApplicationEndpoint` and is defined on the VLAN. The `ApplicationEndpoint` is the endpoint in terms of application addressing and defines the transport layer configuration. The IP-address that is connected to the transport layer is defined by the referenced `NetworkEndpoint`.

[constr_5061] EthernetCommunicationConnectors and referencing SocketAddresses shall be in the same VLAN [Each `EthernetCommunicationConnector` that is referenced by a `SocketAddress` in the role `connector` or `multi-castConnector` shall be referenced by the same `EthernetPhysicalChannel` that aggregates the `SoAdConfig` that in turn aggregates the `SocketAddress`.]()

[constr_3299] SocketAddress.pathMtuDiscoveryEnabled setting dependency [`SocketAddress.pathMtuDiscoveryEnabled` shall only be set to TRUE if `EthernetCommunicationConnector.pathMtuEnabled == TRUE`.]()

[constr_3311] Usage of SocketAddress.flowLabel [`SocketAddress.flowLabel` shall only be used if the aggregated `ApplicationEndpoint` refers to a `NetworkEndpoint` with an `Ipv6Configuration`.]()

[TPS_SYST_02140] SocketAddress.udpChecksumHandling default value [If `SocketAddress.udpChecksumHandling` is not used the value `udpChecksumEnabled` shall be assumed.]()

<i>Enumeration</i>	<code>UdpChecksumCalculationEnum</code>
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	This enumeration defines the UDP checksum calculation.
Literal	Description
<code>udpChecksumDisabled</code>	Udp checksum handling shall be disabled Tags: atp.EnumerationLiteralIndex=1
<code>udpChecksumEnabled</code>	Udp checksum handling shall be enabled Tags: atp.EnumerationLiteralIndex=0

Table 6.127: UdpChecksumCalculationEnum

[TPS_SYST_02141] Semantics of udpChecksumHandling [The semantics of `udpChecksumHandling` is different for the sending and the receiving side:

TX - calculation of UDP checksum:

- `udpChecksumEnabled` means that the UDP checksum is calculated on the transmission side.
- `udpChecksumDisabled` means that the UDP checksum is not calculated but set to zero on the transmission side.

RX - handling of UDP checksum of zero:

- `udpChecksumEnabled` means that the UDP checksum of zero is treated as invalid checksum on receiver side (causing the UDP datagram to be dropped by the receiver). A valid non-zero checksum is accepted and the UDP datagram is forwarded to the upper layer.
- `udpChecksumDisabled` means the the UDP checksum of zero is treated as valid checksum on the receiver side (causing the UDP datagram to be forwarded to the upper layer). A valid non-zero checksum is accepted and the UDP datagram is forwarded to the upper layer as well.

]()

[TPS_SYST_02142] Reception of invalid checksum [On Rx side an invalid checksum should always cause the related UDP datagram to be discarded independent of the `udpChecksumHandling` value.]()

To enable the IPv6 packet filtering the attribute `allowedIPv6ExtHeaders` allows to define a white list of IPv6 Extension Headers that are allowed for a `SocketAddress`. Lists of IPv6 Extension Headers can be defined with the `IPv6ExtHeaderFilterList` element and can be collected in `IPv6ExtHeaderFilterSets`.

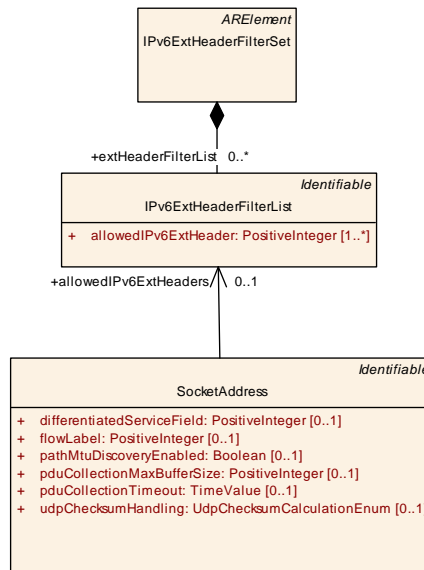


Figure 6.36: IPv6 Extension Header Filter Set

Class	IPv6ExtHeaderFilterSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::IPv6HeaderFilterList			
Note	Set of IPv6 Extension Header Filters. Tags: atp.recommendedPackage=IPv6ExtHeaderFilterSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
extHeaderFilterList	IPv6ExtHeaderFilterList	*	aggr	In order to permit or deny certain types of IPv6 extension headers a white list of IPv6 extension headers can be configured.

Table 6.128: IPv6ExtHeaderFilterSet

Class	IPv6ExtHeaderFilterList			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::IPv6HeaderFilterList			
Note	White list for the filtering of IPv6 extension headers.			
Base	ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
allowedIPv6ExtHeader	PositiveInteger	1..*	attr	IPv6 Extension Header type allowed by this filter.

Table 6.129: IPv6ExtHeaderFilterList

[constr_3276] Prohibition of usage of `allowedIPv6ExtHeaders` in IPv4 `SocketAddress` [IPv4 `SocketAddress` shall not define `allowedIPv6ExtHeaders`. An IPv4 `SocketAddress` aggregates an `ApplicationEndpoint` that refers to a `NetworkEndpoint` that has an `Ipv4Configuration` as `networkEndpointAddress`.]()

[constr_3277] Restriction of usage of `IPv6ExtHeaderFilterLists` in IPv6 `SocketAddress` [All `SocketAddresses` related to the same IPv6 `NetworkEndpoint` shall all reference either no or exactly the same `IPv6ExtHeaderFilterList` with the `allowedIPv6ExtHeaders` attribute.]()

To enable the filtering of Tcp options the attribute `allowedTcpOptions` defines a white list of Tcp options that are allowed for a `SocketAddress`. Lists of Tcp Option filters can be defined with the `TcpOptionFilterList` element and can be collected in `TcpOptionFilterSets`.

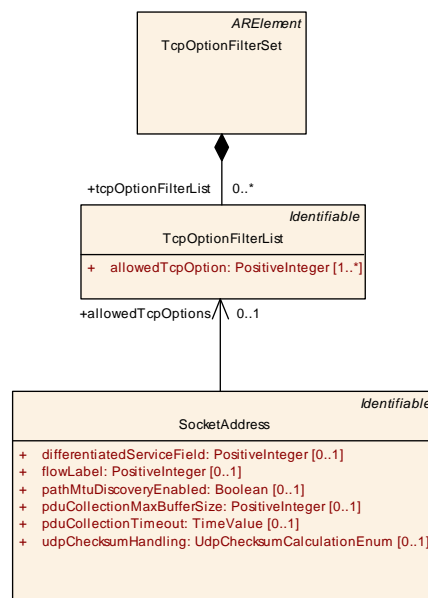


Figure 6.37: Tcp Option Filter Set

Class	TcpOptionFilterSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::TcpOptionFilterSet			
Note	Set of TcpOptionFilterLists. Tags: atp.recommendedPackage=TcpOptionFilterSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
tcpOptionFilterList	TcpOptionFilterList	*	aggr	Collection of white lists for the filtering of TCP options.

Table 6.130: TcpOptionFilterSet

Class	TcpOptionFilterList			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::TcpOptionFilterSet			
Note	White list for the filtering of TCP options.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
allowedTcpOption	PositiveInteger	1..*	attr	TCP option kind allowed by this filter.

Table 6.131: TcpOptionFilterList

[constr_3297] Prohibition of usage of [allowedTcpOptions](#) in Udp [SocketAddress](#) [Udp [SocketAddress](#) shall not define [allowedTcpOptions](#). A Udp [SocketAddress](#) aggregates an [ApplicationEndpoint](#) that has a UdpTp defined as [tpConfiguration](#).]()

6.7.5.1 ApplicationEndpoint

Class	ApplicationEndpoint			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
consumedServiceInstance	ConsumedServiceInstance	*	aggr	Consumed service instances. Tags: atp.Status=obsolete
maxNumberOfConnections	PositiveInteger	0..1	attr	This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.
networkEndpoint	NetworkEndpoint	1	ref	Reference to the network address.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.





Class	ApplicationEndpoint			
providedService Instance	ProvidedService Instance	*	aggr	Provided service instances. Tags: atp.Status=obsolete
tlsCrypto Mapping	TlsCryptoService Mapping	0..1	ref	This reference identifies the applicable TlsCryptoService Mapping that adds the ability for TLS-based encryption on the enclosing ApplicationEndpoint. Tags: atp.Status=draft
tpConfiguration	TransportProtocol Configuration	0..1	aggr	Configuration of the used transport protocol.

Table 6.132: ApplicationEndpoint

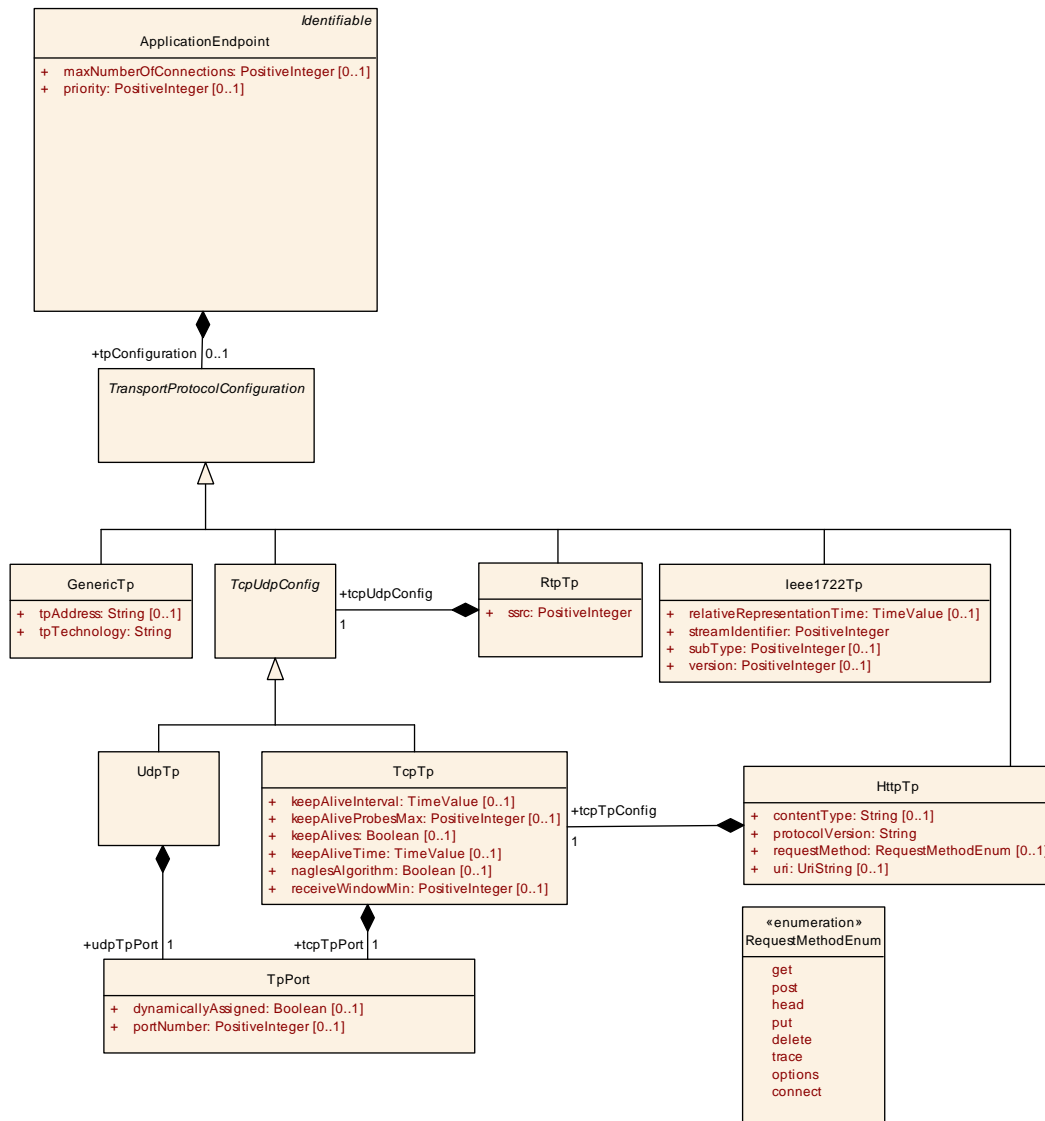


Figure 6.38: Application Endpoint

Class	<i>TransportProtocolConfiguration</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Transport Protocol configuration.			
Base	<i>ARObject</i>			
Subclasses	GenericTp , HttpTp , Ieee1722Tp , RtpTp , TcpUdpConfig			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.133: TransportProtocolConfiguration

The following Transport Protocols are supported by the System Template:

Class	GenericTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for a generic transport protocol.			
Base	<i>ARObject</i> , TransportProtocolConfiguration			
Attribute	Type	Mult.	Kind	Note
tpAddress	String	0..1	attr	Transport Protocol dependent Address.
tpTechnology	String	1	attr	Name of the used Transport Protocol.

Table 6.134: GenericTp

Class	<i>TcpUdpConfig</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Tcp or Udp Transport Protocol Configuration.			
Base	<i>ARObject</i> , TransportProtocolConfiguration			
Subclasses	TcpTp , UdpTp			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.135: TcpUdpConfig

Class	UdpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for UDP configuration.			
Base	<i>ARObject</i> , TcpUdpConfig , TransportProtocolConfiguration			
Attribute	Type	Mult.	Kind	Note
udpTpPort	TpPort	1	aggr	Udp Port configuration.

Table 6.136: UdpTp

Class	TcpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for TCP configuration.			
Base	<i>ARObject</i> , <i>TcpUdpConfig</i> , <i>TransportProtocolConfiguration</i>			
Attribute	Type	Mult.	Kind	Note
keepAliveInterval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
keepAliveProbesMax	PositiveInteger	0..1	attr	Maximum number of times that TCP retransmits an individual data segment before aborting the connection.
keepAlives	Boolean	0..1	attr	Indicates if Keep-Alive messages are send.
keepAliveTime	TimeValue	0..1	attr	Specifies the time in seconds between the last data packet sent and the first keepalive probe.
naglesAlgorithm	Boolean	0..1	attr	Indicates if Nagle's Algorithm is used.
receiveWindowMin	PositiveInteger	0..1	attr	Minimum size of the TCP receive window in byte.
tcpTpPort	TpPort	1	aggr	TCP Port configuration.

Table 6.137: TcpTp

Class	RtpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	RTP over UDP or over TCP as transport protocol.			
Base	<i>ARObject</i> , <i>TransportProtocolConfiguration</i>			
Attribute	Type	Mult.	Kind	Note
ssrc	PositiveInteger	1	attr	Synchronization source identifier uniquely identifies the source of a stream. The synchronization sources within the same RTP session will be unique.
tcpUdpConfig	TcpUdpConfig	1	aggr	Tcp or Udp Configuration.

Table 6.138: RtpTp

Class	Ieee1722Tp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for IEEE 1722 configuration.			
Base	<i>ARObject</i> , <i>TransportProtocolConfiguration</i>			
Attribute	Type	Mult.	Kind	Note
relativeRepresentationTime	TimeValue	0..1	attr	Defines the time when content shall be presented (in seconds). The actual absolute time is creation time plus relative presentation time.
streamIdentifier	PositiveInteger	1	attr	IEEE 1722 stream identifier
subType	PositiveInteger	0..1	attr	Protocol type.
version	PositiveInteger	0..1	attr	Revision of Ieee1722 standard

Table 6.139: Ieee1722Tp

Class	HttpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Http over TCP as transport protocol.			
Base	<i>ARObject</i> , <i>TransportProtocolConfiguration</i>			
Attribute	Type	Mult.	Kind	Note
contentType	String	0..1	attr	Descriptor for the transported content.
protocolVersion	String	1	attr	HTTP Protocol version (e.g. 1.1)
requestMethod	RequestMethodEnum	0..1	attr	HTTP request method to be used.
tcpTpConfig	TcpTp	1	aggr	TcpTp Configuration.
uri	UriString	0..1	attr	URI to be called.

Table 6.140: HttpTp

Class	TpPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Dynamic or direct assignment of a PortNumber.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dynamically Assigned	Boolean	0..1	attr	Indicates whether the source port is dynamically assigned. Tags: atp.Status=obsolete
portNumber	PositiveInteger	0..1	attr	Port Number.

Table 6.141: TpPort

[TPS_SYST_02215] Usage of portNumber with value 0 [The setting of the portNumber to 0 means that the portNumber is assigned dynamically at runtime.]() = 0.

There are different use cases for the usage of portNumber value 0. This setting can be used to describe that the remotePort is dynamically assigned and will be set by the Service Discovery. The localPort can also be set to the value 0 to define that TcpIp need to select an ephemeral port for communication.

[TPS_SYST_01131] TCP/IP and UDP/IP communication over Ethernet [The System Template supports the description of TCP/IP and UDP/IP communication over Ethernet.](RS_SYST_00039)

[TPS_SYST_01089] ApplicationEndpoint priority [The priority at the ApplicationEndpoint shall be used as Ethernet Header information together with the vlanIdentifier. If defined the priority overwrites the defaultPriority that is defined in the VlanMembership and the priority that is defined at the NetworkEndpoint.](RS_SYST_00039)

6.7.5.2 NetworkEndpoint

Class	NetworkEndpoint			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address). Tags: atp.ManifestKind=MachineManifest			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
fullyQualifiedDomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.
infrastructureServices	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed.
ipSecConfig	IPSecConfig	0..1	aggr	Optional IPSec configuration that provides security services for IP packets.
networkEndpointAddress	NetworkEndpointAddress	1..*	aggr	Definition of a Network Address. Tags: xml.name Plural=NETWORK-ENDPOINT-ADDRESSES
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.

Table 6.142: NetworkEndpoint

The [NetworkEndpoint](#) defines the network addressing. The network endpoint may have a priority and a FQDN (Fully Qualified Domain Name) that is used for the Service Discovery (e.g. some.example.host.).

[TPS_SYST_01090] valid [NetworkEndpoint](#) [To build a valid [NetworkEndpoint](#) at least one of the following options shall be defined in the role [NetworkEndpoint](#).
[networkEndpointAddress](#):

- a [MacMulticastConfiguration](#) with a reference to a [MacMulticastGroup](#)
- [Ipv4Configuration](#)
- [Ipv6Configuration](#)

]([RS_SYST_00039](#))

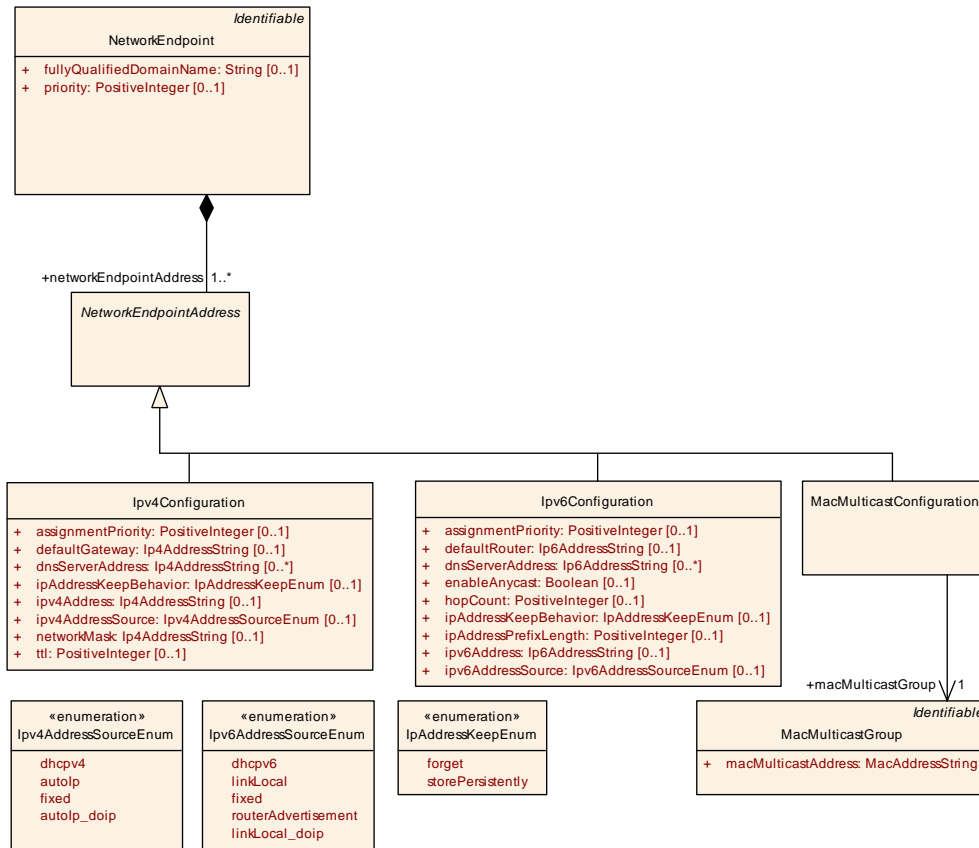


Figure 6.39: Network Endpoint

The attribute `NetworkEndpoint.networkEndpointAddress` defines whether an **IPv4**, **IPv6** or **MAC multicast** address is assigned to the `NetworkEndpoint`.

The reference of the `MacMulticastConfiguration.macMulticastGroup` defines the mapping of IP multicast to MAC multicast.

[TPS_SYST_01088] NetworkEndpoint priority [The `priority` at the `NetworkEndpoint` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership`.] (*RS_SYST_00039*)

Class	<i>NetworkEndpointAddress</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.			
Base	<i>ARObject</i>			
Subclasses	<i>Ipv4Configuration</i> , <i>Ipv6Configuration</i> , <i>MacMulticastConfiguration</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.143: NetworkEndpointAddress

Class	Ipv4Configuration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 4 (IPv4) configuration.			
Base	ARObject, NetworkEndpointAddress			
Attribute	Type	Mult.	Kind	Note
assignmentPriority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServerAddress	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. Tags: xml.namePlural=DNS-SERVER-ADDRESSES
ipAddressKeepBehavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipv4Address	Ip4AddressString	0..1	attr	IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv4AddressSource	Ip4AddressSourceEnum	0..1	attr	Defines how the node obtains its IP address.
networkMask	Ip4AddressString	0..1	attr	Network mask. Notation 255.255.255.255
ttl	PositiveInteger	0..1	attr	Lifespan of data (0..255). The purpose of the TimeToLive field is to avoid a situation in which an undeliverable datagram keeps circulating on a system.

Table 6.144: Ipv4Configuration

Enumeration	Ip4AddressSourceEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines how the node obtains its IPv4-Address.
Literal	Description
autolp	AutoIP is used to dynamically assign IP addresses at device startup. Tags: atp.EnumerationLiteralIndex=0
autolp_doip	Linklocal IPv4 Address Assignment using DoIP Parameters Tags: atp.EnumerationLiteralIndex=2
dhcpv4	DHCP is a service for the automatic IP configuration of a client. Tags: atp.EnumerationLiteralIndex=3
fixed	The IP Address shall be declared manually. Tags: atp.EnumerationLiteralIndex=4

Table 6.145: Ip4AddressSourceEnum

Enumeration	IpAddressKeepEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines the behavior after a dynamic IP address has been assigned.
Literal	Description





Enumeration	IpAddressKeepEnum
forget	After a dynamic IP address has been assigned just use it for this session. Tags: atp.EnumerationLiteralIndex=0
storePersistently	After a dynamic IP address has been assigned store the address persistently. Tags: atp.EnumerationLiteralIndex=1

Table 6.146: IpAddressKeepEnum

Class	Ipv6Configuration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	ARObject, NetworkEndpointAddress			
Attribute	Type	Mult.	Kind	Note
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultRouter	Ip6AddressString	0..1	attr	IP address of the default router.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of pre configured DNS servers. Tags: xml.namePlural=DNS-SERVER-ADDRESSES
enableAnycast	Boolean	0..1	attr	This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).
hopCount	PositiveInteger	0..1	attr	The distance between two hosts. The hop count n means that n gateways separate the source host from the destination host (Range 0..255)
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipAddressPrefix Length	PositiveInteger	0..1	attr	IPv6 prefix length defines the part of the IPv6 address that is the network prefix.
ipv6Address	Ip6AddressString	0..1	attr	IPv6 Address. Notation: FFFF:::FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv6Address Source	Ipv6AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.

Table 6.147: Ipv6Configuration

[TPS_SYST_03002] **Keep behavior of DHCP clients** [The attribute `IpAddressKeepEnum` defines for the DHCP client to either

- persistently store an assigned IP address (`storePersistently`) after it has been fetched, or
- learn it after each start-up (`forget`).

] (`RS_SYST_00052`)

[constr_3298] **Ipv6Configuration.ipv6Address range in case of enableAnycast** [If `Ipv6Configuration.enableAnycast` is set to true then the `Ipv6Configuration.ipv6Address` needs to be in the unicast addressing range.] ()

Enumeration	Ipv6AddressSourceEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines how the node obtains its IPv6-Address.
Literal	Description
dhcpv6	DHCP is a service for the automatic IP configuration of a client. Tags: atp.EnumerationLiteralIndex=0
fixed	The IP Address shall be declared manually. Tags: atp.EnumerationLiteralIndex=1
linkLocal	LinkLocal is intended only for communications within the segment of a local network (a link) or a point-to-point connection that a host is connected to. Tags: atp.EnumerationLiteralIndex=2
linkLocal_doip	Linklocal IPv6 Address Assignment using DoIP Parameters Tags: atp.EnumerationLiteralIndex=3
router Advertisement	IPv6 Stateless Autoconfiguration. Tags: atp.EnumerationLiteralIndex=4

Table 6.148: Ipv6AddressSourceEnum

Class	MacMulticastConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	References a per cluster globally defined MAC-Multicast-Group.			
Base	<i>ARObject</i> , <i>NetworkEndpointAddress</i>			
Attribute	Type	Mult.	Kind	Note
macMulticast Group	MacMulticastGroup	1	ref	Reference to a macMulticastGroup.

Table 6.149: MacMulticastConfiguration

Primitive	Ip4AddressString
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	This is used to specify an IP4 address. Notation: 255.255.255.255 Tags: xml.xsd.customType=IP4-ADDRESS-STRING xml.xsd.pattern=(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) ANY xml.xsd.type=string

Table 6.150: Ip4AddressString

Primitive	Ip6AddressString
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes





Primitive	Ip6AddressString
Note	<p>This is used to specify an IP6 address. Notation: FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF</p> <p>Alternative notations, short-cuts with duplicate colons like ::, etc. or mixtures using colons and dots, are not allowed.</p> <p>Tags: xml.xsd.customType=IP6-ADDRESS-STRING xml.xsd.pattern=[0-9A-Fa-f]{1,4}(:[0-9A-Fa-f]{1,4}){7,7} ANY xml.xsd.type=string</p>

Table 6.151: Ip6AddressString

In addition infrastructure services may be provided or consumed by the [NetworkEndpoints](#).

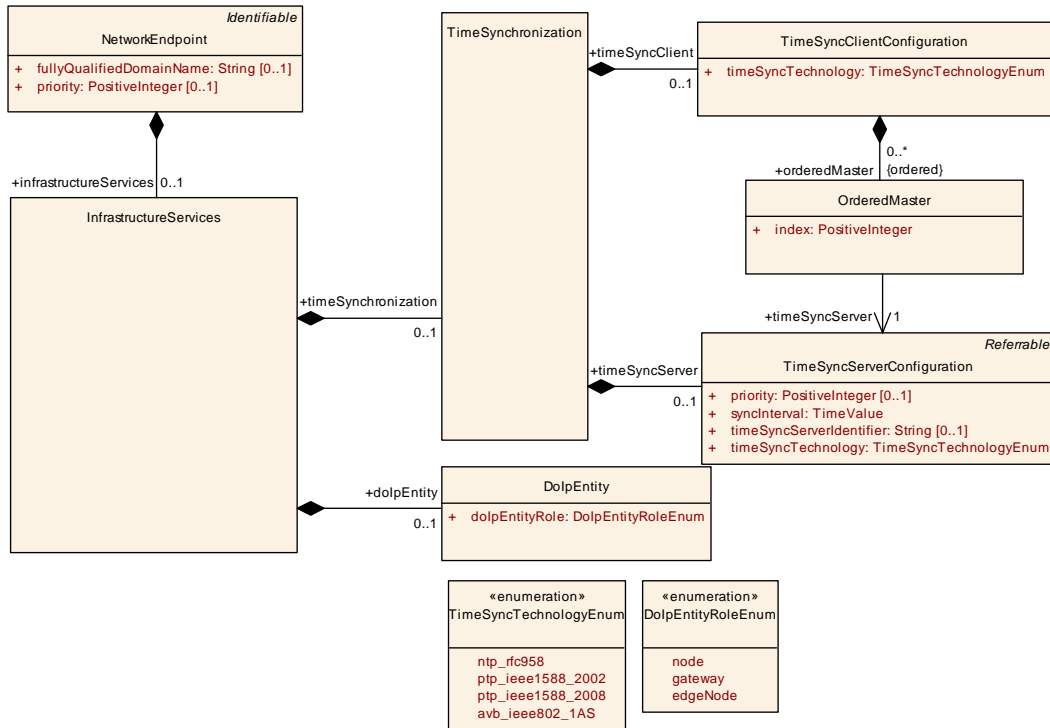


Figure 6.40: Network Endpoint Infrastructure Services

Class	InfrastructureServices			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the network infrastructure services provided or consumed.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
dolpEntity	DolpEntity	0..1	aggr	Defines whether a infrastructure service that runs on the network endpoint is a DoLP-Entity.





Class	InfrastructureServices			
time Synchronization	TimeSynchronization	0..1	aggr	Defines the servers / clients in a time synchronised network.

Table 6.152: InfrastructureServices

The [TimeSyncServerConfiguration](#) provides a time synchronization service.

[constr_3257] TimeSyncTechnology of servers and clients in a time synchronized network. [[TimeSyncClientConfiguration.timeSyncTechnology](#) shall have the same value as the [TimeSyncServerConfiguration.timeSyncTechnology](#) that is referenced in the [TimeSyncClientConfiguration.orderedMaster](#) list.]()

Please note that there may be several [timeSyncServers](#) defined in the [TimeSyncClientConfiguration.orderedMaster](#) list, but only one is accepted at runtime. In case that a master is not available any more a master transition will be processed according to the defined [TimeSyncClientConfiguration.orderedMaster](#) list. The next defined [timeSyncServer](#) in the [OrderedMaster](#) list will take over the master functionality.

Class	TimeSynchronization			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the servers / clients in a time synchronised network.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
timeSyncClient	TimeSyncClientConfiguration	0..1	aggr	Configuration of the time synchronisation client.
timeSyncServer	TimeSyncServerConfiguration	0..1	aggr	Configuration of the time synchronisation server.

Table 6.153: TimeSynchronization

Class	TimeSyncClientConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation client.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
orderedMaster (ordered)	OrderedMaster	*	aggr	Defines a list of ordered NetworkEndpoints. Tags: xml.namePlural=ORDERED-MASTER-LIST
timeSync Technology	TimeSyncTechnologyEnum	1	attr	Defines the time synchronisation technology used.

Table 6.154: TimeSyncClientConfiguration

Class	TimeSyncServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation server.			
Base	<i>ARObject</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
priority	PositiveInteger	0..1	attr	Server Priority.
syncInterval	TimeValue	1	attr	Synchronisation interval used by the time synchronisation server (in seconds).
timeSyncServer Identifier	String	0..1	attr	Identifier of the TimeSyncServer.
timeSync Technology	TimeSyncTechnology Enum	1	attr	Defines the time synchronisation technology used. Possible values are: NTP_RFC958, PTP_IEEE1588_2002, PTP_IEEE1588_2008, AVB_IEEE802_1AS and others.

Table 6.155: TimeSyncServerConfiguration

Class	OrderedMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Element in the network endpoint list.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
index	PositiveInteger	1	attr	Defines the order of the network endpoint list (e.g. 0, 1, 2, ...).
timeSyncServer	TimeSyncServer Configuration	1	ref	Reference to a master (Time Sync Server).

Table 6.156: OrderedMaster

Enumeration	TimeSyncTechnologyEnum		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology		
Note	Timesynchronization. Server/Client configuration.		
Literal	Description		
avb_ieee802_1AS	Ethernet AVB compliant IEEE802.1AS Precision Time Protocol Tags: atp.EnumerationLiteralIndex=0		
ntp_rfc958	Network Time Protocol (NTP) Tags: atp.EnumerationLiteralIndex=1		
ptp_ieee1588_2002	Precision Time Protocol (PTP) IEEE 1588-2002 Tags: atp.EnumerationLiteralIndex=2		
ptp_ieee1588_2008	Precision Time Protocol (PTP) IEEE 1588-2008 Tags: atp.EnumerationLiteralIndex=3		

Table 6.157: TimeSyncTechnologyEnum

The [DoIpEntity](#) (Diagnostics over Internet Protocol, ISO 13400) defines the [DoIp](#) role this [NetworkEndpoint](#) has.

Class	DolpEntity			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	ECU providing this infrastructure service is a DoIP-Entity.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dolpEntityRole	DolpEntityRoleEnum	1	attr	Identifies the role in terms of DoIP this network-node has.

Table 6.158: DolpEntity

Enumeration	DolpEntityRoleEnum		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology		
Note	DoIP role a network-node has.		
Literal	Description		
edgeNode	Network node is a DoIP gateway that accepts external connections. Tags: atp.EnumerationLiteralIndex=0		
gateway	Network node is a Gateway between the DoIP network and other networks. Tags: atp.EnumerationLiteralIndex=1		
node	Network node is a Dolp node. Tags: atp.EnumerationLiteralIndex=2		

Table 6.159: DolpEntityRoleEnum

6.7.5.3 SOME/IP Service Instances

The AUTOSAR protocol SOME/IP is a Service Discovery protocol that is used to communicate the availability of functional entities called services in the in-vehicle communication and to control the send behavior of event messages to receivers that subscribed to receive these events (Publish/Subscribe). A Service may also provide methods that a client is able to call (Request/Response).

Please note that the AUTOSAR Classic Platform does not support Service Interfaces. To mimic a Service Interface in the classic platform any combination of [ClientServerInterfaces](#), [SenderReceiverInterfaces](#) or [TriggerInterfaces](#) may be used to describe a service to which later a SOME/IP Service Id is assigned.

The assignment of SOME/IP Service Ids is done on the Service Instance level in the System Description. An [AbstractServiceInstance](#) collected in the [ServiceInstanceCollectionSet](#) is describing such a Service Instance.

Class	ServiceInstanceCollectionSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Collection of ServiceInstances Tags: atp.recommendedPackage=ServiceInstanceCollectionSets			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
serviceInstance	AbstractServiceInstance	*	aggr	ServiceInstances that are part of the collection. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.160: ServiceInstanceCollectionSet

Class	AbstractServiceInstance (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Provided and Consumed Ethernet Service Instances that are available at the ApplicationEndpoint.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Subclasses	ConsumedServiceInstance , ProvidedServiceInstance			
Attribute	Type	Mult.	Kind	Note
capabilityRecord	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.
methodActivationRoutingGroup	PduActivationRoutingGroup	0..1	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing from and to TCP/IP-sockets. Tags: atp.Status=obsolete

Table 6.161: AbstractServiceInstance

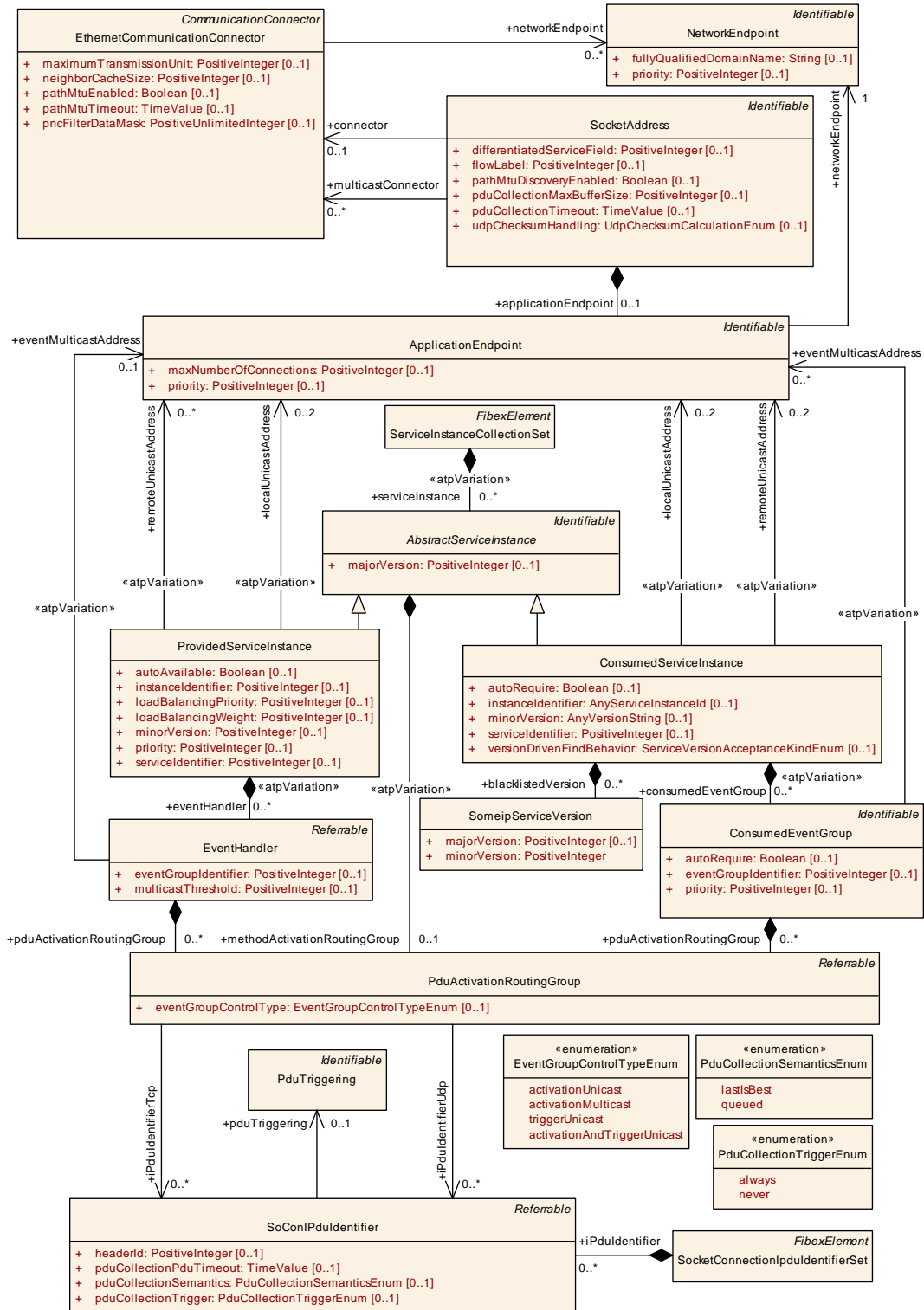


Figure 6.41: Provided and Consumed Service Instances

SOME/IP allows for the specification of additional information about the `AbstractServiceInstance` with the Capability Record that allows to transport arbitrary configuration strings (key/value pairs). This allows to encode additional information like the name of a service or its configuration.

[TPS_SYST_02216] Configuration of **capabilityRecords** [A Capability Record (key/value pair) is configurable with the `AbstractServiceInstance.capabilityRecord` and the two attributes `key` and `value`.]()

Class	TagWithOptionalValue			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::TagWithOptionalValue			
Note	A tagged value is a combination of a tag (key) and a value that gives supplementary information that is attached to a model element. Please note that keys without a value are allowed. Tags: atp.ManifestKind=ServiceInstanceManifest			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
key	String	1	attr	Defines a key.
value	String	0..1	attr	Defines the corresponding value.

Table 6.162: TagWithOptionalValue

[TPS_SYST_01094] allowed **key/value** `TagWithOptionalValue` combinations [The following `key/value` combinations are supported:

- `key` present, with no `value` (e.g. "passreq" -> password required for this service)
- `key` present, with empty `value` (e.g. "PlugIns=" -> server supports plugins, but none are presently installed)
- `key` present, with non-empty `value` (e.g. "PlugIns=JPEG,MPEG2,MPEG4")

] (*RS_SYST_00039*)

The following chapters are describing the configuration of SOME/IP `ProvidedServiceInstances` and `ConsumedServiceInstances` in more detail. Please note that currently the communication between a `ProvidedServiceInstance` and a `ConsumedServiceInstance` is restricted to a VLAN (see also [`constr_5079`]).

6.7.5.4 ProvidedServiceInstance

The `ProvidedServiceInstance` is used to define a Service provider.

[TPS_SYST_02217] **SOME/IP Service offer** [The `EcuInstance` on which the `ProvidedServiceInstance` is deployed offers the Service Instance over SOME/IP with the `serviceIdentifier` and `instanceIdentifier`. The version of the Service that is offered is described by the attributes `majorVersion` and `minorVersion`.] (*RS_SYST_00039*)

[`constr_5062`] **SOME/IP `ProvidedServiceInstances` of the same serviceInterface on one `EcuInstance`** [Different `ProvidedServiceInstances` with the same `serviceIdentifier` and the same `majorVersion` and different `instanceIdentifiers` shall not be mapped to the same UDP/TCP port number and IP address combination that is represented by referenced `ApplicationEndpoint` and its referenced `NetworkEndpoint`.]()

The reason for this restriction is that the Instance IDs are only used for Service Discovery but are not contained in the SOME/IP header. So if for example two `ProvidedServiceInstances` of the same ServiceInterface are provided on the same `EcuInstance` and a client wants to call a method of one of these `ProvidedServiceInstances` the only possibility for the client to distinguish the `ProvidedServiceInstances` is the port number over which the individual `ProvidedServiceInstances` are provided.

[constr_5063] `ProvidedServiceInstance.serviceIdentifier` is mandatory [The `ProvidedServiceInstance.serviceIdentifier` is mandatory.]()

[constr_5064] `ProvidedServiceInstance.majorVersion` is mandatory [The `ProvidedServiceInstance.majorVersion` is mandatory.]()

[constr_5065] `ProvidedServiceInstance.minorVersion` is mandatory [The `ProvidedServiceInstance.minorVersion` is mandatory.]()

[constr_5066] `ProvidedServiceInstance.instanceIdentifier` is mandatory [The `ProvidedServiceInstance.instanceIdentifier` is mandatory.]()

[constr_5067] `ProvidedServiceInstance` shall be unique in respect of `serviceIdentifier`, `instanceIdentifier`, `majorVersion` [On a VLAN each `ProvidedServiceInstance` shall have a different `serviceIdentifier`, `instanceIdentifier` and `majorVersion` value combination.]()

The reason for this constraint is that the Service Discovery messages have to be unambiguous on the VLAN.

In other words no two `ProvidedServiceInstances` in a variant bound model shall have the same `serviceIdentifier`, `instanceIdentifier` and `majorVersion` value combination.

Class	ProvidedServiceInstance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	<i>ARObject</i> , <i>AbstractServiceInstance</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
autoAvailable	Boolean	0..1	attr	Defines that this <code>ProvidedServiceInstance</code> shall be offered by the service discovery at ECU start.
eventHandler	EventHandler	*	aggr	Collection of event groups provided by the <code>ProvidedServiceInstance</code> Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
instanceIdentifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.
loadBalancingPriority	PositiveInteger	0..1	attr	Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.





Class	ProvidedServiceInstance			
loadBalancingWeight	PositiveInteger	0..1	attr	Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.
localUnicastAddress	ApplicationEndpoint	0..2	ref	The local address over which the PSI is provided (udp, tcp or both). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
minorVersion	PositiveInteger	0..1	attr	Minor Version of the Service that is provided by this ProvidedServiceInstance.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
remoteUnicastAddress	ApplicationEndpoint	*	ref	This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
sdServerConfig	SdServerConfig	0..1	aggr	Service Discovery Server configuration. Tags: atp.Status=obsolete
sdServerTimerConfig	SomeipSdServerServiceInstanceConfig	0..1	ref	Server specific configuration settings relevant for the SOME/IP service discovery. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is offered.

Table 6.163: ProvidedServiceInstance

[TPS_SYST_02218] ProvidedServiceInstance deployment [The deployment of a [ProvidedServiceInstance](#) to an [EcuInstance](#) is realized with the [localUnicastAddress](#) reference. The referenced [ApplicationEndpoint](#) defines the TCP/UDP Port and the [NetworkEndpoint](#) that is referenced by the [ApplicationEndpoint](#) defines the IP Address on which the Service can be reached. This Endpoint information is transported in the SOME/IP Service Offer message to the clients. With the [SocketAddress.connector](#) reference the deployment to an [EcuInstance](#) is achieved.] ([RS_SYST_00039](#))

The AUTOSAR BswM is used to aggregate the availability of all entities which make up a service instance. When all entities are available, the service instance as such is available. When a service instance becomes available the SD Module will usually send an announcement message so other ECUs can learn about the availability and the location (IP address and UDP or TCP Port) of that service instance.

Please note that the Service provider and the Service consumer do not need to find each other if the [ProvidedServiceInstance](#) contains a [remoteUnicastAddress](#) reference to an [ApplicationEndpoint](#). If this reference is set the Server will not get Clients address from the connection request. The server knows the address from the Client from the configuration and all necessary socket connections can be set up from this configuration information. The client still need to subscribe to event groups that are of interest for him in this setup.

[TPS_SYST_02219] Static configuration between `ProvidedServiceInstance` and `ConsumedServiceInstance` [If the `ProvidedServiceInstance` contains a `remoteUnicastAddress` reference to an `ApplicationEndpoint` the SoAd will setup one single `SocketConnection` between the `localUnicastAddress` and the `remoteUnicastAddress`.] (*RS_SYST_00039*)

[TPS_SYST_02220] Maximal number of clients that may connect to the local server address [If the `ProvidedServiceInstance` references an `ApplicationEndpoint` in the role `localUnicastAddress` then the attribute `ApplicationEndpoint.maxNumberOfConnections` defines the maximal number of clients (with a dynamic address) that will be able to connect to this local server address. The SoAd will setup a `SocketConnection` for each potential client with `localUnicastAddress` and ANY remote Address (ANY Port and ANY IP-Address).] (*RS_SYST_00039*)

[constr_5068] `ProvidedServiceInstance.localUnicastAddress` shall be IP Unicast [If defined, the `ProvidedServiceInstance.localUnicastAddress` shall point to an IP Unicast address.] ()

[constr_5069] `ProvidedServiceInstance.remoteUnicastAddress` shall be IP Unicast [The `ProvidedServiceInstance.remoteUnicastAddress` shall point to an IP Unicast address.] ()

In other words the `localUnicastAddress` and `remoteUnicastAddress` are not allowed to point to an IP Multicast address. Please note that an IP Unicast address can be defined as a fixed address or be retrieved via dynamic mechanisms, e.g. DHCP.

Please note that a Service may provide some portions (Events, Methods) over TCP and other portions over UDP. This is the reason why the `ProvidedServiceInstance` is able to reference up to two `localUnicastAddresses`.

[TPS_SYST_02221] `ProvidedServiceInstance.localUnicastAddress` reference target [The `ProvidedServiceInstance` is allowed to have:

- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP Port,
- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP Port,
- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP Port and a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP Port.

] (*RS_SYST_00039*)

[constr_3379] Multiple `SocketAddress` entries with the same IP Address, Protocol and Port in the context of a given `EcuInstance` [If there are two or more `SocketAddress` entities within the scope of one `SoAdConfig` in the scope of one `EcuInstance` that have the same static (fixed at configuration time) IP Address, Protocol and Port in the aggregated `ApplicationEndpoint` and `NetworkEndpoint`,

(e.g., 192.168.1.1, Tcp and 10000, respectively) then only one of these `SocketAddress` elements shall be referenced by `ProvidedServiceInstances/ConsumedServiceInstances` in the role `localUnicastAddress.()`

Rationale for [constr_3379]: There can be only one representation of the `ProvidedServiceInstance/ConsumedServiceInstance` using the given IP Address, Protocol and Port in the Sd module configuration in the context of a given `EcuInstance`. Therefore, defining `ProvidedServiceInstance/ConsumedServiceInstance` and assign it to several `ApplicationEndpoints` would in this case require a merge of potentially different attribute values of the `ProvidedServiceInstances` and/or `ConsumedServiceInstances` in the System Description and such situation is avoided by this constraint.

Class	PduActivationRoutingGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Group of Pdus that can be activated or deactivated for transmission over a socket connection.			
Base	<i>ARObject, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
eventGroupControlType	EventGroupControlTypeEnum	0..1	attr	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.
iPduIdentifierTcp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Tcp in case that the referencing PduActivationRoutingGroup is activated.
iPduIdentifierUdp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Udp in case that the referencing PduActivationRoutingGroup is activated.

Table 6.164: PduActivationRoutingGroup

Enumeration	EventGroupControlTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	Types of a RoutingGroups for the event communication.
Literal	Description
activationAndTriggerUnicast	Activate the data path for unicast events and triggered unicast events that are sent out after a client got subscribed. Tags: atp.EnumerationLiteralIndex=0
activationMulticast	Activate the data path for multicast events of an EventGroup. Tags: atp.EnumerationLiteralIndex=1
activationUnicast	Activate the data path for unicast events of an EventGroup. Tags: atp.EnumerationLiteralIndex=2





Enumeration	EventGroupControlTypeEnum
triggerUnicast	Activate the data path for triggered unicast events that are sent out after a client got subscribed. Tags: atp.EnumerationLiteralIndex=3

Table 6.165: EventGroupControlTypeEnum

Class	SoConIPduIdentifier			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Identification of Pdu content on a socket connection. This Identifier is required in case that multiple Pdus are transmitted over the same socket connection.			
Base	ARObject, Referrable			
Attribute	Type	Mult.	Kind	Note
headerId	PositiveInteger	0..1	attr	If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.
pduCollection PduTimeout	TimeValue	0..1	attr	Defines the timeout in seconds the PDU collection shall be transmitted at the latest after this PDU has been put into the buffer.
pduCollection Semantics	PduCollection SemanticsEnum	0..1	attr	Specifies if the referenced PduTriggering shall be collected using a queued (i.e. all PDU instances) or last-is-best (i.e. only the last PDU instance) semantics. If this attribute is not present the behavior of "queued" is assumed.
pduCollection Trigger	PduCollectionTrigger Enum	0..1	attr	Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.
pduTriggering	PduTriggering	0..1	ref	Reference to a Pdu that is transmitted over a socket connection.

Table 6.166: SoConIPduIdentifier

Enumeration	PduCollectionSemanticsEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	Defines the collection semantics for the PDU collection feature.
Literal	Description
lastIsBest	Only the latest PDU instances are transmitted. Tags: atp.EnumerationLiteralIndex=0
queued	All instances of PDUs are transmitted. Tags: atp.EnumerationLiteralIndex=1

Table 6.167: PduCollectionSemanticsEnum

[TPS_SYST_02222] Usage of **headerId** [If multiple **SoConIPduIdentifiers** are referenced by **PduActivationRoutingGroup** in the role **iPduIdentifierTcp** or in the role **iPduIdentifierUdp** then the **headerId** information shall be used to distinguish between the different **Pdus**.] (**RS_SYST_00039**)

[constr_5070] **headerIds** of **ProvidedServiceInstances** shall be unique on a **SocketAddress** per communication direction [Each **SoConIPduIdentifier**

that is referenced in the context of a `ProvidedServiceInstance` (including aggregated `EventHandler`) that in turn belongs to a collection of `ProvidedServiceInstances` that point to the same `ApplicationEndpoint` shall have a unique `headerId` defined in the group of all `SoConIPduIdentifiers` defined for transmission on this `ApplicationEndpoint` per communication direction. |()

Please note that a Method Call and a Method Return may use the same `headerId` for identification since the communication direction is different in this case.

[constr_3322] Consistent setting of `SoConIPduIdentifier.pduCollectionSemantics` in the context of one `SocketAddress` [The value of the attribute `SoConIPduIdentifier.pduCollectionSemantics` shall be identical for all referenced `SoConIPduIdentifiers` within the context of a given `SocketAddress`.] |()

[TPS_SYST_02223] Activation/Deactivation of `PduActivationRoutingGroups` [The routing of `Pdus` to and from a socket may be activated or deactivated with a `PduActivationRoutingGroup` depending on the availability of `AbstractServiceInstances`, `EventHandlers` or `ConsumedEventGroups` that send or receive the data.] (*RS_SYST_00039*)

The Routing Group Activation Table is controlled in AUTOSAR by the Service Discovery module.

[TPS_SYST_02224] Methods provided by a `ProvidedServiceInstance` [If the `ProvidedServiceInstance` is offered by the Service Discovery protocol then the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` is activated. All Methods that are provided by the `ProvidedServiceInstance` can be called by the interested clients and the server will respond to these calls. If the `ProvidedServiceInstance` offer is stopped then the `PduActivationRoutingGroup` is deactivated.] (*RS_SYST_00039*)

Please note that according to [TPS_SYST_02151] the relationship between the call and return is achieved by means of meta data items attached to the `Pdus` by the Socket Adapter.

[TPS_SYST_02081] `PduTriggering` that is used for ClientServer Communication [A `PduTriggering` that points to an `ISignalIPdu` that aggregates an `ISignalToIPduMapping` that in turn references an `ISignal` that refers to a `ClientServerToSignalMapping.callSignal` or to `ClientServerToSignalMapping.returnSignal` is designated as `PduTriggering` that is used for ClientServer Communication.] |()

[TPS_SYST_02225] Service methods provided over UDP [Method `Pdus` (Call and Return `ISignalIPdus`) of a `ProvidedServiceInstance` that are offered for access over UDP are described by `SoConIPduIdentifiers` referenced by the `ipduIdentifierUdp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` by the `ProvidedServiceInstance`.] (*RS_SYST_00039*)

[TPS_SYST_02226] **Service methods provided over TCP** [Method Pdus (Call and Return `ISignalIPdus`) of a `ProvidedServiceInstance` that are offered for access over TCP are described by `SoConIPduIdentifiers` referenced by the `ipduIdentifierTcp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` by the `ProvidedServiceInstance`.] ([RS_SYST_00039](#))

[TPS_SYST_02227] **Publishing of a SOME/IP Event group** [A `ProvidedServiceInstance` publishes an event group for each `EventHandler` that is aggregated in the role `eventHandler`. The `eventGroupIdentifier` identifies the SOME/IP Event Group in the context of the `ProvidedServiceInstance`. With the publishing of an event group the server offers to push notifications about updates to all clients that are subscribed to the event group.] ([RS_SYST_00039](#))

Class		EventHandler		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This element represents an event group as part of the Provided Service Instance.			
Base	<code>ARObject</code> , <code>Referrable</code>			
Attribute	Type	Mult.	Kind	Note
consumedEventGroup	<code>ConsumedEventGroup</code>	*	ref	All consumers of the event are referenced here. Tags: atp.Status=obsolete
eventGroupIdentifier	PositiveInteger	0..1	attr	Unique Identifier that identifies the EventGroup in SOME/IP. This Identifier is sent as Eventgroup ID in SOME/IP Service Discovery messages.
eventMulticastAddress	<code>ApplicationEndpoint</code>	0..1	ref	Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the <code>multicastThreshold</code> is exceeded. This address is transmitted in the SD-SubscribeEventGroupAck Message to client (answer to SD-SubscribeEventGroup). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
multicastThreshold	PositiveInteger	0..1	attr	Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast. If configured to 0 only unicast will be used. If configured to 1 the first client will be already served by multicast. If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast. This does not influence the handling of initial events, which are served using unicast only.
pduActivationRoutingGroup	<code>PduActivationRoutingGroup</code>	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.
routingGroup	<code>SoAdRoutingGroup</code>	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events. Tags: atp.Status=obsolete
sdServerConfig	<code>SdServerConfig</code>	0..1	aggr	Server configuration parameter for Service-Discovery. Tags: atp.Status=obsolete





Class	EventHandler			
sdServerEg TimingConfig	SomeipSdServerEventGroupTimingConfig	0..1	ref	Server Timing configuration settings that are EventGroup specific. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.168: EventHandler

[TPS_SYST_02228] Transmission of events over UDP/TCP Port [The events of an event group that are described by an [EventHandler](#) are transmitted either:

- via IP Unicast on the UDP/TCP Port that is defined by the [ApplicationEndpoint](#) that is referenced by the [ProvidedServiceInstance](#) in the role [localUnicastAddress](#) if the [eventMulticastAddress](#) is not set or the [multicastThreshold](#) is not reached, or
- via IP Multicast on the local UDP/TCP Port that is defined by the [ApplicationEndpoint](#) that is referenced by the [ProvidedServiceInstance](#) in the role [localUnicastAddress](#) to the IP Multicast remote Address that is defined by the [ApplicationEndpoint](#) that is referenced by the [EventHandler](#) in the role [eventMulticastAddress](#) if the [multicastThreshold](#) is reached.

] ([RS_SYST_00039](#))

[TPS_SYST_02180] Usage of [EventHandler.multicastThreshold](#) [The switching between IP-Unicast and IP-Multicast is guided by the server with the [EventHandler.multicastThreshold](#) attribute and by the number of subscribed clients to the [EventHandler](#).

The Server will change the transmission of events to Multicast if the [multicastThreshold](#) of the corresponding [EventHandler](#) is reached by the number of subscribed clients. If the number of subscribed clients is smaller then the configured [multicastThreshold](#), the transmission of events takes place via unicast communication.] ()

[constr_5071] [EventHandler.eventMulticastAddress](#) reference target [The [ApplicationEndpoint](#) that is referenced by an [EventHandler](#) in the role [eventMulticastAddress](#) shall reference a [NetworkEndpoint](#) that defines an IP Multicast Address.] ()

[constr_5072] [EventHandler](#) without defined [eventMulticastAddress](#) [If an [EventHandler](#) that is aggregated by a [ProvidedServiceInstance](#) does not have a defined [eventMulticastAddress](#) then the [multicastThreshold](#) shall be set to the value 0 (IP Unicast only).] ()

[TPS_SYST_02229] Event groups provided by a [ProvidedServiceInstance](#) [If the [ProvidedServiceInstance](#) publishes an event group then the [PduActivationRoutingGroup](#) that is aggregated in the role [pduActivationRoutingGroup](#) by an [EventHandler](#) is activated. The interested clients can subscribe to the published event groups until the offer is stopped.] ([RS_SYST_00039](#))

[TPS_SYST_02230] PduActivationRoutingGroups for event groups [The `PduActivationRoutingGroup` that is aggregated by the `EventHandler` in the role `pduActivationRoutingGroup` enables the routing of a group of `Pdus` that are related with the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifiers`.

- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` enables the routing over IP Unicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` enables the routing over IP Multicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `triggerUnicast` enables the routing of initial events that are sent out by the server immediately after a client got subscribed. This `PduActivationRoutingGroup` can be used for SOME/IP Field notifiers.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationAndTriggerUnicast` enables the routing over IP Unicast and makes sure that initial events are sent out by the server immediately after a client got subscribed. This `PduActivationRoutingGroup` can be used for SOME/IP Field notifiers.

]()

[TPS_SYST_02231] PduActivationRoutingGroups for methods [The `PduActivationRoutingGroup` enables the routing of a group of `Pdus` that are related with the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifiers`. The `eventGroupControlType` attribute is irrelevant for the `PduActivationRoutingGroup` that is aggregated by the `AbstractServiceInstance` in the role `methodActivationRoutingGroup`.]()

[constr_5073] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is aggregated by an EventHandler [An `EventHandler` that aggregates a `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` or `triggerUnicast` or `activationAndTriggerUnicast` shall be aggregated by a `ProvidedServiceInstance` that has a `localUnicastAddress` reference that points to an IP Unicast Address.]()

[constr_5074] PduActivationRoutingGroup with eventGroupControlType set to activationMulticast that is aggregated by an EventHandler [An `EventHandler` that aggregates a `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` shall have an `eventMulticastAddress` reference that points to a “remote”

IP Multicast Address. The `ProvidedServiceInstance` that aggregates the `EventHandler` shall have a `localUnicastAddress` reference to a “local” UDP `ApplicationEndpoint`.]()

[constr_5075] Allowed references of `SoConIPduIdentifiers` by `PduActivationRoutingGroup` with `eventGroupControlType` set to `activationMulticast` and allowed `SoConIPduIdentifier` references [A `PduActivationRoutingGroup` with `eventGroupControlType` set to `activationMulticast` is allowed to reference `SoConIPduIdentifiers` only in the `iPduIdentifierUdp` role.]()
()

[TPS_SYST_02232] Events provided over UDP [Pdus of an `EventHandler` that are provided over UDP are described by `SoConIPduIdentifiers` referenced by the `iPduIdentifierUdp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by the `EventHandler`.] (*RS_SYST_00039*)

[TPS_SYST_02233] Events provided over TCP [Pdus of an `EventHandler` that are provided over TCP are described by `SoConIPduIdentifiers` referenced by the `iPduIdentifierTcp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by the `EventHandler`.] (*RS_SYST_00039*)

[constr_5076] `PduActivationRoutingGroup` with `iPduIdentifierTcp` reference that is aggregated by a `ProvidedServiceInstance` [If the `PduActivationRoutingGroup` contains the `iPduIdentifierTcp` reference then the aggregating `ProvidedServiceInstance` shall contain a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP address.]()
()

[constr_5077] `PduActivationRoutingGroup` with `iPduIdentifierUdp` reference that is aggregated by a `ProvidedServiceInstance` [If the `PduActivationRoutingGroup` contains the `iPduIdentifierUdp` reference then the aggregating `ProvidedServiceInstance` shall contain a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP address.]()
()

[constr_5078] `PduTriggerings` referenced by a `PduActivationRoutingGroup` shall be on the same VLAN as the referencing `PduActivationRoutingGroup` [Each `PduTriggering` referenced by a `PduActivationRoutingGroup` via `SoConIPduIdentifier` shall be aggregated by the same VLAN (`EthernetPhysicalChannel`) to which the `AbstractServiceInstance` that aggregates the `PduActivationRoutingGroup` belongs via the `localUnicastAddress`.]()
()

[constr_5079] Service communication is restricted to one VLAN [All `SocketAddress` elements that are referenced by a `AbstractServiceInstance` with the `localUnicastAddress` and `remoteUnicastAddress` shall belong to the same VLAN (`EthernetPhysicalChannel`).]()
()

[constr_5080] `ApplicationEndpoints` referenced by `EventHandlers` and by the aggregating `ProvidedServiceInstance` shall be in the same VLAN [The

`ApplicationEndpoint` that is referenced by an `EventHandler` in the role `event-MulticastAddress` shall belong to the same VLAN (`EthernetPhysicalChannel`) as the `ApplicationEndpoint` that is referenced by the `localUnicastAddress` reference from the `ProvidedServiceInstance` that aggregates the `EventHandler`.]()

[constr_3456] Existence of `ProvidedServiceInstance.loadBalancingPriority` and `ProvidedServiceInstance.loadBalancingWeight` [The attributes `ProvidedServiceInstance.loadBalancingPriority` and `ProvidedServiceInstance.loadBalancingWeight` shall either not exist or be defined both.]()

[TPS_SYST_01108] `ProvidedServiceInstance` priority [The `priority` in the `ProvidedServiceInstance` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership`, the `priority` that is defined at the `NetworkEndpoint` and the `priority` that is defined at the `ApplicationEndpoint`.]
(*RS_SYST_00039*)

method (named `Method1`) and two events (named `Event1` and `Event2`). In addition the service contains two different event groups that are represented by two `EventHandler`s `EG1` and `EG2`.

On VFB level the method is represented by a `ClientServerOperation` and each event by a `VariableDataPrototype` in a `SenderReceiverInterface`.

The `ClientServerOperation` is mapped by a `ClientServerToSignalMapping` to a `callSignal` and to a `returnSignal` that in turn are mapped via `ISignals` into an `ISignalIPdu`. Each `VariableDataPrototype` is mapped by the `SenderReceiverToSignalMapping` or by the `SenderReceiverToSignalGroupMapping` to one or several `SystemSignals` that in turn are mapped via `ISignals` into an `ISignalIPdu`.

The `PduTriggerings` of these `ISignalIPdu`s are referenced by the `SoConIPduIdentifier` that assigns a `headerId` to each `ISignalIPdu`.

The event group `EG2` contains `Event1` and `Event2` and the `PduActivationRoutingGroup` named `EG2_PSI_25_1` activates the `SoConIPduIdentifiers` that are representing these two events for transmission over IP Unicast since the `eventGroupControlType` is set to `activationUnicast`.

The event group `EG1` contains `Event1` and the `PduActivationRoutingGroup` named `EG1_PSI_25_1` activates the `SoConIPduIdentifier` for transmission over IP Multicast since the `eventGroupControlType` is set to `activationMulticast`. `EventHandler` `EG1` has the `multicastThreshold` attribute set to 1. This means that the first client that will subscribe to this event group will be served via IP Multicast. This is the reason why the `EventHandler` `EG1` points with the `eventMulticastAddress` to an IP Multicast address to which the event will be transmitted.

In the `SoAd` configuration such a `System Description` will result in a `SoAdSocketConnectionGroup` with a `SoAdSocketLocalPort` and `SoAdSocketLocalAddressRef` that are derived from the `ApplicationEndpoint` referenced in the `localUnicastAddress` role by the `ProvidedServiceInstance`.

The `SoAdSocketConnectionGroup` will contain three `SoAdSocketConnections` since the `maxNumberOfConnections` is set to 3 in this example. The `ProvidedServiceInstance` does not contain a `remoteUnicastAddress` reference. This means that `Service Discovery` is used and that the `SoAdSocketRemoteIpAddress` needs to be configured to `ANY` and the `SoAdSocketRemotePort` to 0 in all three `SoAdSocketConnections`.

In addition one multicast `SoAdSocketConnection` will be added to the `SoAdSocketConnectionGroup` since the `EventHandler` `EG1` points with the `eventMulticastAddress` to a multicast address.

6.7.5.5 ConsumedServiceInstance

The `ConsumedServiceInstance` is used to define a SOME/IP Service consumer. Please note that in the AUTOSAR model all necessary information about the searched service is configurable on the client side. A model that contains only the `ConsumedServiceInstance` description is sufficient for the configuration of the `EcuInstance` to which the `ConsumedServiceInstance` is deployed. This is different to former versions of the System Template. The design criterion for the model was to disentangle the `ProvidedServiceInstance` and the `ConsumedServiceInstance` from each other.

[TPS_SYST_02234] SOME/IP Service search [A defined `ConsumedServiceInstance` is searching for a SOME/IP Service Instance that fulfills all of the following conditions:

- Service Identifier that matches the value set in `serviceIdentifier`,
- Service Instance Identifier that matches the value set in `instanceIdentifier` or ANY Service Instance of the Service in case the `instanceIdentifier` is set to ANY.
- Service major version that matches the value set in `majorVersion`,
- Service major version:
 - in case `versionDrivenFindBehavior = exactOrAnyMinorVersion`: Service minor version that matches the value set in `minorVersion` or ANY minor version of the Service Instance in case the `minorVersion` is set to ANY
 - in case `versionDrivenFindBehavior = minimumMinorVersion`: Service minor version that matches at least the value set in `minorVersion` or is higher

](*RS_SYST_00039*)

[constr_5081] `ConsumedServiceInstance.serviceIdentifier` is mandatory [The `ConsumedServiceInstance.serviceIdentifier` is mandatory.]()

[constr_5082] `ConsumedServiceInstance.majorVersion` is mandatory [The `ConsumedServiceInstance.majorVersion` is mandatory.]()

[constr_5083] `ConsumedServiceInstance.minorVersion` is mandatory [The `ConsumedServiceInstance.minorVersion` is mandatory.]()

[constr_3560]{DRAFT} `minimumMinorVersion` and `ConsumedServiceInstance.minorVersion` value [The `ConsumedServiceInstance.minorVersion` shall not have the value ANY if `versionDrivenFindBehavior = minimumMinorVersion`.]()

[constr_5084] `ConsumedServiceInstance.instanceIdentifier` is mandatory [The `ConsumedServiceInstance.instanceIdentifier` is mandatory.]()

Class		ConsumedServiceInstance		
Package		M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances		
Note		Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.		
Base		ARObject, AbstractServiceInstance, Identifiable, MultilanguageReferrable, Referrable		
Attribute	Type	Mult.	Kind	Note
autoRequire	Boolean	0..1	attr	Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.
blacklistedVersion	SomeipServiceVersion	*	aggr	Collection of blacklisted versions. Tags: atp.Status=draft
consumedEventGroup	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
instanceIdentifier	AnyServiceInstanceId	0..1	attr	This attribute represents the ability to describe the required service instance ID.
localUnicastAddress	ApplicationEndpoint	0..2	ref	The local address over which the CSI is consumed (udp, tcp or both). Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
minorVersion	AnyVersionString	0..1	attr	Minor Version of the ServiceInterface. Value can be set to a number that represents the Minor Version of the searched service or to ANY.
providedServiceInstance	ProvidedServiceInstance	0..1	ref	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedServiceInstance. Tags: atp.Status=obsolete
remoteUnicastAddress	ApplicationEndpoint	0..2	ref	This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
sdClientConfig	SdClientConfig	0..1	aggr	Service Discovery Client configuration. Tags: atp.Status=obsolete
sdClientTimerConfig	SomeipSdClientServiceInstanceConfig	0..1	ref	Client specific configuration settings relevant for the SOME/IP service discovery. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is searched.
versionDrivenFindBehavior	ServiceVersionAcceptanceKindEnum	0..1	attr	Defines the service discovery find behavior. Tags: atp.Status=draft

Table 6.169: ConsumedServiceInstance

[TPS_SYST_03050]{DRAFT} Usage of **ConsumedServiceInstance.blacklistedVersion** [A service connection of a **ConsumedServiceInstance** to a **ProvidedServiceInstance** is not considered for service discovery if the **SomeipServiceVersion.minorVersion** exists in the collection of **SomeipServiceVersions** aggregated at the **ConsumedServiceInstance** in the role **blacklistedVersion**.]

()

A typical scenario for using a blacklist may be: For a certain `ConsumedServiceInstance` a certain compatible provider service version inside a system may not work which may have been identified after the design phase. In order to keep the system running this certain provider version won't be considered in the service search if it has been blacklisted. Therefore, the `ConsumedServiceInstance` may connect only to `ProvidedServiceInstances` that fulfill the search criteria and are not blacklisted.

[constr_3559]{DRAFT} `ConsumedServiceInstance.blacklistedVersion` is restricted to the usage of `minorVersion` [The `majorVersion` attribute shall not be used in the `SomeipServiceVersion` that is aggregated by the `ConsumedServiceInstance` in the role `blacklistedVersion`.]()

[TPS_SYST_02235] `ConsumedServiceInstance` deployment [The deployment of a `ConsumedServiceInstance` to an `EcuInstance` is realized with the `localUnicastAddress` reference. The referenced `ApplicationEndpoint` defines the TCP/UDP Port and the `NetworkEndpoint` that is referenced by the `ApplicationEndpoint` defines the IP Address on which the service consumer is located. Over the defined TCP/UDP port the service consumer sends out the method calls to a service provider that was found by the Service Discovery search. The endpoint information is also transported in the SOME/IP `SubscribeEventGroup` message to the server to indicate on which address the incoming events are expected. With the `SocketAddress.connector` reference the deployment to an `EcuInstance` is achieved.] (*RS_SYST_00039*)

Please note that the Service provider and the Service consumer do not need to find each other if the `ConsumedServiceInstance` contains a `remoteUnicastAddress` reference to an `ApplicationEndpoint`. If this reference is set the client will not get server's address from the Service Offer message. The client already knows the address from the service provider from the configuration and all necessary socket connections can be set up from this configuration information. The client still needs to subscribe to event groups that are of interest for him in this setup.

[TPS_SYST_02236] Static configuration between `ConsumedServiceInstance` and `ProvidedServiceInstance` [If the `ConsumedServiceInstance` contains a `remoteUnicastAddress` reference to an `ApplicationEndpoint` the SoAd will setup one single `SocketConnection` between the `localUnicastAddress` and the `remoteUnicastAddress`.] (*RS_SYST_00039*)

[TPS_SYST_02237] Maximal number of servers that may connect to the local client address [If the `ConsumedServiceInstance` references an `ApplicationEndpoint` in the role `localUnicastAddress` and the `remoteUnicastAddress` is not defined then the attribute `ApplicationEndpoint.maxNumberOfConnections` defines the maximal number of remote endpoints that will be able to connect to this local client address. The SoAd will setup a `SocketConnection` for each potential remote endpoint. The local Address of the `SocketConnection` is derived from `ConsumedServiceInstance.localUnicastAddress`. The remote Address is set to ANY address in case that the `remoteUnicastAddress` is not used.] (*RS_SYST_00039*)

[constr_5085] ConsumedServiceInstance.localUnicastAddress shall be IP Unicast [If defined, the `ConsumedServiceInstance.localUnicastAddress` shall point to an IP Unicast address.](/)

In other words the `localUnicastAddress` is not allowed to point to a IP Multicast address. Please note that the `ConsumedServiceInstance.localUnicastAddress` does not need to be set in all cases as specified in [TPS_SYST_02238].

[constr_5086] ConsumedServiceInstance.remoteUnicastAddress shall be IP Unicast [The `ConsumedServiceInstance.remoteUnicastAddress` shall point to an IP Unicast address.](/)

Please note that a Service may provide some portions (Events, Methods) over TCP and and other portions over UDP. This is the reason why the `ConsumedServiceInstance` is able to reference up to two `localUnicastAddresses`.

[TPS_SYST_02238] ConsumedServiceInstance.localUnicastAddress reference target [The `ConsumedServiceInstance` is allowed to have:

- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP Port,
- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP Port,
- a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP Port and a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP Port.
- no configured `localUnicastAddress`.

](RS_SYST_00039)

The Routing Group Activation is controlled in AUTOSAR by the Service Discovery module.

[TPS_SYST_02239] Methods consumed by a ConsumedServiceInstance [If the `ConsumedServiceInstance` finds the searched service then the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` is activated. All Methods that are provided by the found `ProvidedServiceInstance` can be called by `ConsumedServiceInstance`.](RS_SYST_00039)

Please note that according to [TPS_SYST_02151] the relationship between the call and return is achieved by means of meta data items attached to the `Pdus` by the Socket Adapter.

[TPS_SYST_02240] Service methods consumed over UDP [Method `Pdus` (Call and Return `ISignalIPdus`) of a `ConsumedServiceInstance` that are accessed over UDP are described by `SoConIPduIdentifiers` referenced by the `ipduIdentifierUdp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` by the `ConsumedServiceInstance`.](RS_SYST_00039)

[TPS_SYST_02241] **Service methods consumed over TCP** [Method Pdus (Call and Return `ISignalIPdus`) of a `ConsumedServiceInstance` that are accessed over TCP are described by `SoConIPduIdentifiers` referenced by the `iPduIdentifierTcp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` by the `ConsumedServiceInstance`.] ([RS_SYST_00039](#))

[TPS_SYST_02242] **Subscription to a SOME/IP Event group** [A `ConsumedServiceInstance` subscribes to a published event group via a `ConsumedEventGroup` that is aggregated in the role `consumedEventGroup`. The `eventGroupIdentifier` identifies the SOME/IP Event Group in the context of the `ConsumedServiceInstance` to which the consumer subscribes.] ([RS_SYST_00039](#))

Class	ConsumedEventGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This element represents an event-group to which the service consumer wants to subscribe.			
Base	<code>ARObject</code> , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
application Endpoint	ApplicationEndpoint	0..1	ref	Defines the application endpoint where the events of the event group are received in case of multicast reception. Tags: atp.Status=obsolete
autoRequire	Boolean	0..1	attr	Defines that this <code>ConsumedEventGroup</code> shall be requested (subscribed) as soon as the corresponding <code>ConsumedServiceInstance</code> is requested. This could be at ECU start, if <code>ConsumedServiceInstance.autoRequire</code> is set to TRUE or as soon as the <code>ConsumedServiceInstance</code> is requested by the application, if <code>ConsumedServiceInstance.autoRequire</code> is set to FALSE.
eventGroup Identifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.
eventMulticast Address	ApplicationEndpoint	*	ref	This reference defines the multicast address or a multicast address resource where the events of the event group are received. If the multicast address is determined via configuration and not at runtime via service discovery this reference points to the multicast address over which the events will be received. If the multicast address is determined at runtime via service discovery this reference shall be used to define the necessary local multicast address resources, i.e. RAM space in the <code>Tcplp</code> module in which the multicast address is stored at runtime. Please note that in this case the referenced address may be defined as ANY UDP port and ANY IP address since the multicast address will be received at runtime. If several multicast addresses are considered to be used the <code>ConsumedEventGroup</code> shall point to different <code>ApplicationEndpoint</code> objects to reserve the necessary resources in the configuration. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The <code>ServiceDiscovery</code> module is able to activate and deactivate the PDU routing for receiving events.





Class	ConsumedEventGroup			
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events. Tags: atp.Status=obsolete
sdClientConfig	SdClientConfig	0..1	aggr	The readiness to receive events is defined by the Service Discovery of the ConsumedEventGroup. The Event Handler shall know about this announcement to decide about the submission of events. Therefore the Event Handler may be configured with Service-Discovery Client attributes. Tags: atp.Status=obsolete
sdClientTimer Config	SomeipSdClientEventGroupTimingConfig	0..1	ref	Client Timing configuration settings that are EventGroup specific. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.170: ConsumedEventGroup

[TPS_SYST_02243] Reception of events over UDP/TCP Port in case of Service Discovery [The events of an event group that are described by an [ConsumedEventGroup](#) are received either:

- via IP Unicast on the UDP/TCP Port that is defined by the [ApplicationEndpoint](#) that is referenced by the [ConsumedServiceInstance](#) in the role [localUnicastAddress](#) or
- via IP Multicast:
 - in case of dynamic SD configuration the Multicast IP Address and UDP Port to which the events are transmitted will be provided by the server at runtime in the SOME/IP SubscribeEventGroupAck message,
 - in case of static configuration the Multicast IP Address and UDP Port to which the events are transmitted shall be configured by [ConsumedEventGroup.eventMulticastAddress](#).

]([RS_SYST_00039](#))

Please note that the [ConsumedEventGroup.eventMulticastAddress](#) shall also be used in case of a dynamic SD Configuration where the client learns the multicast address at runtime. In this case the [ApplicationEndpoint](#) and the corresponding [NetworkEndpoint](#) that are referenced by [eventMulticastAddress](#) define a resource in the Tcplp module where the multicast address that will be determined at runtime will be stored. Since the referenced [ApplicationEndpoint](#) and the corresponding [NetworkEndpoint](#) are placeholders the configured TP Port and IP address shall be set to ANY.

There are different scenarios that are considered:

- The same multicast address is used by all potential servers. On the client a resource for this single multicast address needs to be reserved. All `ConsumedEventGroups` need to reference the same placeholder `ApplicationEndpoint` with `eventMulticastAddress`. In the Ecu Configuration one `TcpIpLocalAddr` container is created from the placeholder `ApplicationEndpoint`. In addition one `SoAdSocketConnectionGroup` is created that points with the `SoAdSocketLocalAddressRef` to the `TcpIpLocalAddr` container. All Pdus received over multicast are assigned to this `SoAdSocketConnectionGroup`.
- The server is using different multicast addresses, e.g. Multicast Address A for `EventHandler` A and Multicast Address B for `EventHandler` B. On the client side resources for all used multicast addresses need to be reserved. In the example the `ConsumedEventGroup` A needs to reference one placeholder `ApplicationEndpoint` with `eventMulticastAddress`. The `ConsumedEventGroup` B needs to reference a second placeholder `ApplicationEndpoint` with `eventMulticastAddress`. In the Ecu Configuration a `TcpIpLocalAddr` container is created for each used placeholder. In addition one `SoAdSocketConnectionGroup` per placeholder is created. In the example the Pdus of `ConsumedEventGroup` A are assigned to one `SoAdSocketConnectionGroup` and Pdus of `ConsumedEventGroup` B are assigned to the second `SoAdSocketConnectionGroup`. With this approach it is possible to define at configuration time which Pdus share the same multicast address, and which ones are going over different multicast addresses.

If more than one multicast address is used it is not known at configuration time which server will publish which address at which point in time. The configuration needs to be prepared for all possible scenarios. Please be aware that this comes with a high resource consumption.

[constr_3262] `ConsumedEventGroup.eventGroupIdentifier` is mandatory
[The `ConsumedEventGroup.eventGroupIdentifier` is mandatory.]()

[constr_3457] Uniqueness of `ConsumedEventGroup.eventGroupIdentifier` in the scope of a `ConsumedServiceInstance` [Each `ConsumedEventGroup` that is aggregated by a `ConsumedServiceInstance` shall have a unique `eventGroupIdentifier` value in the scope of the aggregating `ConsumedServiceInstance`.]
()

[TPS_SYST_02244] `ConsumedServiceInstance` without a defined `localUnicastAddress` [If a `ConsumedServiceInstance` does not have a defined `localUnicastAddress` then the events will be received over IP Multicast only.](*RS_SYST_00039*)

[TPS_SYST_02245] Event groups consumed by a `ConsumedServiceInstance` [If the `ConsumedServiceInstance` subscribes to an event group then the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by an `ConsumedEventGroup` is activated.](*RS_SYST_00039*)

[TPS_SYST_02246] PduActivationRoutingGroups for ConsumedEventGroups [The `PduActivationRoutingGroup` that is aggregated by the `ConsumedEventGroup` in the role `pduActivationRoutingGroup` enables the routing of a group of `Pdus` that are related to the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifiers`.

- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` enables the receiving of events over IP Unicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` enables the receiving of events over IP Multicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `triggerUnicast` enables the receiving of initial events that are sent out by the server immediately after the client got subscribed. This `PduActivationRoutingGroup` can be used for SOME/IP Field notifiers.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationAndTriggerUnicast` enables the routing over IP Unicast and makes sure that initial events are sent out by the server immediately after the client got subscribed. This `PduActivationRoutingGroup` can be used for SOME/IP Field notifiers.

]()

[constr_5087] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is referenced by a ConsumedEventGroup [A `ConsumedEventGroup` that aggregates a `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` or `triggerUnicast` or `activationAndTriggerUnicast` shall be aggregated by a `ConsumedServiceInstance` that has a `localUnicastAddress` reference that points to an IP Unicast Address.]()

[TPS_SYST_02247] Events consumed over UDP [Pdus of an `ConsumedEventGroup` that are consumed over UDP are described by `SoConIPduIdentifiers` referenced by the `iPduIdentifierUdp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by the `ConsumedEventGroup`.] (*RS_SYST_00039*)

[TPS_SYST_02248] Events consumed over TCP [Pdus of an `ConsumedEventGroup` that are consumed over TCP are described by `SoConIPduIdentifiers` referenced by the `iPduIdentifierTcp` reference from the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by the `ConsumedEventGroup`.] (*RS_SYST_00039*)

[constr_5088] PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ConsumedServiceInstance [If the PduActivationRoutingGroup contains the iPduIdentifierTcp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP address.]()

[constr_5089] PduActivationRoutingGroup with iPduIdentifierUdp reference that is aggregated by a ConsumedServiceInstance [If the PduActivationRoutingGroup contains the iPduIdentifierUdp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP address.]()

[TPS_SYST_02014] ConsumedEventGroup priority [The priority in the ConsumedEventGroup shall be used as Ethernet Header information together with the vlanIdentifier. If defined the priority overwrites the defaultPriority that is defined in the VlanMembership, the priority that is defined at the NetworkEndpoint and the priority that is defined at the ApplicationEndpoint.]()

[constr_5090] ApplicationEndpoints referenced by ConsumedEventGroups and by the aggregating ConsumedServiceInstance shall be in the same VLAN [The ApplicationEndpoint that is referenced by an ConsumedEventGroup in the role eventMulticastAddress shall belong to the same VLAN (EthernetPhysicalChannel) as the ApplicationEndpoint that is referenced by the localUnicastAddress reference from the ConsumedServiceInstance that aggregates the ConsumedEventGroup.]()

The event group EG2 contains Event1 and Event2 and the `PduActivationRoutingGroup` named EG2_CSI_25_1 activates the `SoConIPduIdentifiers` that are representing these two events for transmission over IP Unicast since the `eventGroupControlType` is set to `activationUnicast`.

The event group EG1 contains Event1 and the `PduActivationRoutingGroup` named EG1_CSI_25_1 activates the `SoConIPduIdentifier` for transmission over IP Multicast since the `eventGroupControlType` is set to `activationMulticast`.

In the SoAd configuration such a System Description will result in a `SoAdSocketConnectionGroup` with a `SoAdSocketLocalPort` and `SoAdSocketLocalAddressRef` that are derived from the `ApplicationEndpoint` referenced in the `localUnicastAddress` role by the `ConsumedServiceInstance`.

The `SoAdSocketConnectionGroup` will contain one single `SoAdSocketConnection` since the `maxNumberOfConnections` is not set in this example. The `ProvidedServiceInstance` does not contain a `remoteUnicastAddress` reference. This means that Service Discovery is used and that the `SoAdSocketRemoteIpAddress` needs to be configured to ANY and the `SoAdSocketRemotePort` to 0 in the `SoAdSocketConnection`.

Since only a single multicast placeholder `ApplicationEndpoint` is referenced with the `eventMulticastAddress` by all existing `ConsumedEventGroups` one multicast `SoAdSocketConnectionGroup` needs to be created with two `SoAdSocketConnections`.

6.7.5.6 Service Discovery Server Configuration

For every `ProvidedServiceInstance` on a Server different phases are existing where a suitable Service Discovery Message sending behavior is configurable:

- Down
- Available
 - Initial Wait Phase
 - Repetition Phase
 - Main Phase

[TPS_SYST_02249] Service Discovery Message sending behavior on `ProvidedServiceInstance` [The Service Discovery Message sending behavior on a `ProvidedServiceInstance` is configurable with the `SomeipSdServerServiceInstanceConfig` that is referenced in the role `sdServerTimerConfig`.]()

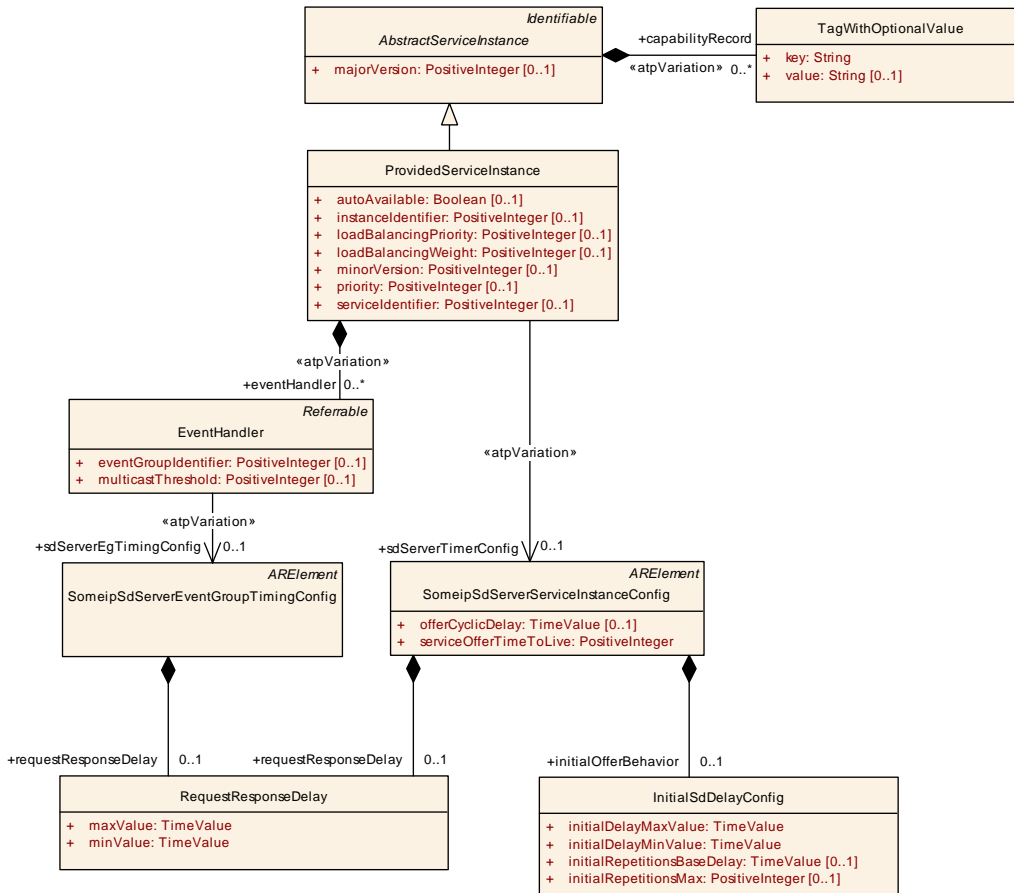


Figure 6.44: Model of SD Server Timing

Class	SomeipSdServerServiceInstanceConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Server specific settings that are relevant for the configuration of SOME/IP Service-Discovery. Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=SomeipSdTimingConfigs			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
initialOfferBehavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server. Tags: atp.Status=draft
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds).
requestResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds. The Service Discovery shall delay answers to entries that were transported in a multicast SOME/IP-SD message (e.g. FindService). Tags: atp.Status=draft





Class	SomeipSdServerServiceInstanceConfig			
serviceOfferTimeToLive	PositiveInteger	1	attr	Defines the time in seconds the service offer is valid.

Table 6.171: SomeipSdServerServiceInstanceConfig

Class	InitialSdDelayConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This element is used to configure the offer behavior of the server and the find behavior on the client. Tags: atp.ManifestKind=ServiceInstanceManifest			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
initialDelayMaxValue	TimeValue	1	attr	Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).
initialDelayMinValue	TimeValue	1	attr	Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig).
initialRepetitionsBaseDelay	TimeValue	0..1	attr	The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay.
initialRepetitionsMax	PositiveInteger	0..1	attr	Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).

Table 6.172: InitialSdDelayConfig

Class	RequestResponseDelay			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Time to wait before answering the query. Tags: atp.ManifestKind=ServiceInstanceManifest			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
maxValue	TimeValue	1	attr	Maximum allowable response delay to entries received by multicast in seconds.
minValue	TimeValue	1	attr	Minimum allowable response delay to entries received by multicast in seconds.

Table 6.173: RequestResponseDelay

[TPS_SYST_02174] Initial Wait Phase configuration for a [ProvidedServiceInstance](#) [The Initial Wait Phase for a [ProvidedServiceInstance](#) is configured with the [initialOfferBehavior](#) and the two attributes [initialDelayMinValue](#) and [initialDelayMaxValue](#).]()

When a calculated random timer based on these min and max values expires, the first `OfferService` message will be sent out.

[TPS_SYST_02258] Shared random timer for [ProvidedServiceInstance](#) service discovery [If several [ProvidedServiceInstances](#) reference the same

`SomeipSdServerServiceInstanceConfig` in the role of `sdServerTimerConfig` and if it is ensured that all `ProvidedServiceInstances` are requested / released in the same point in time (e.g. assigned to the same `ConsumedProvidedServiceInstanceGroup` and referenced only by one `ConsumedProvidedServiceInstanceGroup`, or all referencing `ProvidedServiceInstances` has set `autoAvailable` to TRUE and no runtime changes to the availability is performed), then the timing behavior is shared between the referencing `ProvidedServiceInstances`. Thus, if the calculated random timer based on the min and max values of the `InitialSdDelayConfig` expires, then OfferService entries of all referencing `ProvidedServiceInstances` shall be sent out.]()

[TPS_SYST_02259] Shared random timer for `ProvidedServiceInstance` service discovery [If several `ProvidedServiceInstances` reference the same `SomeipSdServerServiceInstanceConfig` in the role of `sdServerTimerConfig` and if it cannot be ensured that all `ProvidedServiceInstances` are requested / released in the same point in time, then the timing behavior is handled per `ProvidedServiceInstance`. The timing behavior shall be derived from the referenced `SomeipSdServerServiceInstanceConfig` for each referencing `ProvidedServiceInstance`.]()

Note: [TPS_SYST_02259] supports an efficient modeling if timing behaviors are equal for the referencing `ProvidedServiceInstances`.

[TPS_SYST_02260] Individual random timer for `ProvidedServiceInstance` service discovery [If `ProvidedServiceInstances` reference their own `SomeipSdServerServiceInstanceConfig` in the role of `sdServerTimerConfig`, then the timing behavior is handled per `ProvidedServiceInstance`. Thus, every `ProvidedServiceInstance` has its own calculated random timer based on the min and max value of the `InitialSdDelayConfig`. If the calculated random timer expires, then a OfferService entry of the referencing `ProvidedServiceInstance` shall be sent out.]()

Note: [TPS_SYST_02260] enables a scattering of OfferService within the configured min and max value of the `InitialSdDelayConfig` of different `ProvidedServiceInstances`.

[TPS_SYST_02175] Repetition Wait Phase configuration for a `ProvidedServiceInstance` [The Repetition Wait Phase for a `ProvidedServiceInstance` is configured with the `initialOfferBehavior` and the two attributes `initialRepetitionsMax` and `initialRepetitionsBaseDelay`.]()

If the Repetition Phase is entered the Service Discovery waits for the `initialRepetitionsBaseDelay` and transmits an OfferService entry. If the amount of sent OfferService entries reaches `initialRepetitionsMax` the Main Phase will be entered.

If `initialRepetitionsMax` is configured to 0 the Repetition Phase will be skipped and the Main Phase will be entered.

[TPS_SYST_02176] Main Phase configuration for a [ProvidedServiceInstance](#)
 [The Main Phase for a [ProvidedServiceInstance](#) is configured with the [offerCyclicDelay](#) attribute of [SomeipSdServerServiceInstanceConfig](#).]()

The [OfferService](#) entry will be sent cyclically with an interval that is defined by the value of attribute [offerCyclicDelay](#).

[TPS_SYST_02177] TTL for Offer Service Entries [The lifetime of a [ProvidedServiceInstance](#) is configurable with the [serviceOfferTimeToLive](#) attribute of [SomeipSdServerServiceInstanceConfig](#).]()

If the time that is configured by [serviceOfferTimeToLive](#) expires the [ProvidedServiceInstance](#) will no longer be offered.

[TPS_SYST_02178] Servers [RequestResponseDelay](#) for received [FindService](#) entries [The Server will delay the [OfferService](#) answer to a received multicast [FindService](#) entry by the configured [SomeipSdServerServiceInstanceConfig.requestResponseDelay](#).

The actual delay will be randomly chosen between the [maxValue](#) and [minValue](#).]()

The [ProvidedServiceInstance](#) aggregates an [EventHandler](#) in the role [eventHandler](#) that allows to define service instance specific configuration settings for a SOME/IP [EventGroup](#). The [EventGroup](#) specific timing settings are configured in the [SomeipSdServerEventGroupTimingConfig](#) that is referenced by the [EventHandler](#) in the role [sdServerEgTimingConfig](#).

Class	SomeipSdServerEventGroupTimingConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	EventGroup specific timing configuration settings. Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=SomeipSdTimingConfigs			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
requestResponseDelay	RequestResponseDelay	0..1	aggr	The Service Discovery shall delay answers to unicast messages triggered by multicast messages (e.g. Subscribe Eventgroup after Offer Service). Tags: atp.Status=draft

Table 6.174: SomeipSdServerEventGroupTimingConfig

[TPS_SYST_02182] Servers [RequestResponseDelay](#) for received [SubscribeEventGroup](#) entries [The Server will delay the [SubscribeEventGroupAck](#) answer to a received [SubscribeEventGroup](#) message that was triggered by a multicast [ServiceOffer](#) by the configured [SomeipSdServerEventGroupTimingConfig.requestResponseDelay](#) that is referenced by the [EventHandler](#) in the role [sdServerEgTimingConfig](#).

The actual delay will be randomly chosen between the [maxValue](#) and [minValue](#).]()

6.7.5.7 Service Discovery Client Configuration

For every `ConsumedServiceInstance` on a Client different phases are existing:

- Down
- Requested
 - Initial Wait Phase
 - Repetition Phase
 - Main Phase

[TPS_SYST_02250] Service Discovery Message sending behavior on `ConsumedServiceInstance` [The Service Discovery Message sending behavior on a `ConsumedServiceInstance` is configurable with the `SomeipSdClientServiceInstanceConfig` that is referenced in the role `sdClientTimerConfig`.]()

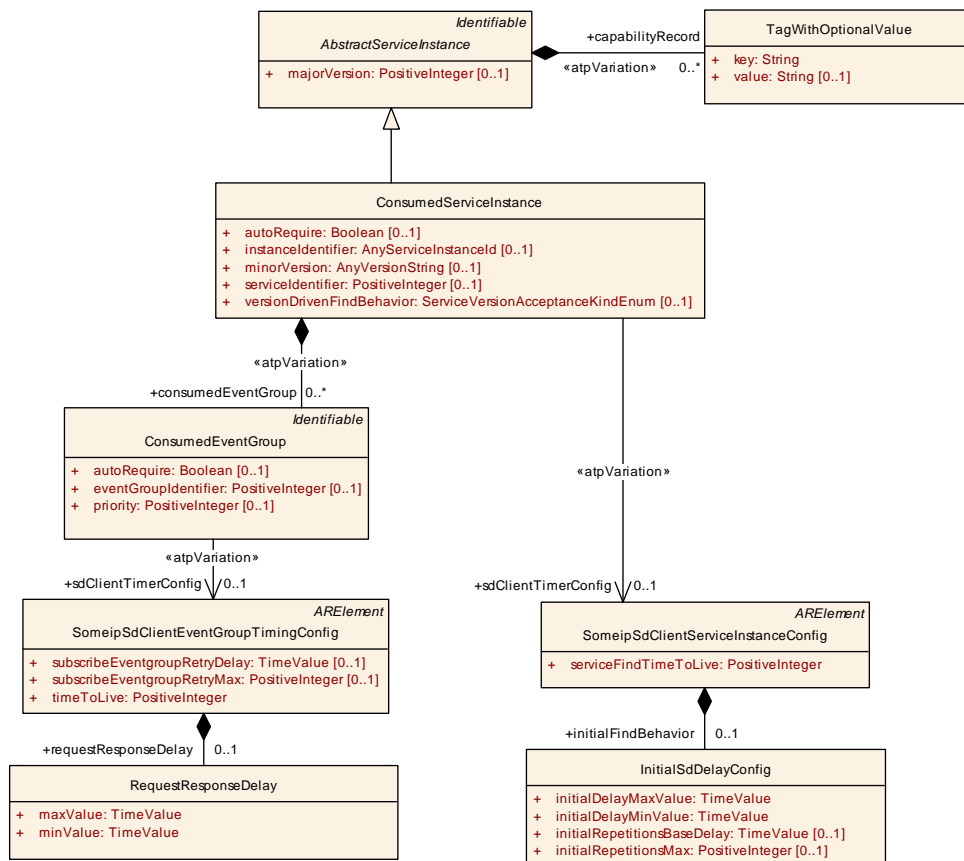


Figure 6.45: Model of SD Client Timing

[TPS_SYST_02183] Initial Wait Phase configuration for a `ConsumedServiceInstance` [The Initial Wait Phase for a `ConsumedServiceInstance` is configured with the `initialFindBehavior` and the two attributes `initialDelayMinValue` and `initialDelayMaxValue`.]()

If a calculated random timer based on these min and max values expires the first `FindService` entry will be sent out. When the calculated random timer expires and no `OfferService` is received the Repetition Phase will be entered.

[TPS_SYST_02261] Shared random timer for `ConsumedServiceInstance` service discovery [If several `ConsumedServiceInstances` reference the same `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig` and if it is ensured that all `ConsumedServiceInstances` are requested / released in the same point in time (e.g. assigned to the same `ConsumedProvidedServiceInstanceGroup` and referenced only by one `ConsumedProvidedServiceInstanceGroup`, or all referencing `ConsumedServiceInstance` has set `autoRequire` to TRUE and no runtime changes to the require state is performed), then the timing behavior is shared between the referencing `ConsumedServiceInstances`. Thus, if the calculated random timer based on the min and max values of the `InitialSdDelayConfig` expires, then `FindService` entries of all referencing `ConsumedServiceInstances` shall be sent out.]()

[TPS_SYST_02262] Shared random timer for `ConsumedServiceInstance` service discovery [If several `ConsumedServiceInstances` reference the same `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig` and if it cannot be ensured that all `ConsumedServiceInstances` are requested / released in the same point in time, then the timing behavior is handled per `ConsumedServiceInstance`. The timing behavior shall be derived from the referenced `SomeipSdClientServiceInstanceConfig` for each referencing `ConsumedServiceInstance`.]()

Note: [TPS_SYST_02262] support an efficient modeling if timing behaviors are equal for the referencing `ConsumedServiceInstances`.

[TPS_SYST_02263] Individual random timer for `ConsumedServiceInstance` service discovery [If `ConsumedServiceInstances` reference their own `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig`, then the timing behavior is handled per `ConsumedServiceInstance`. Thus, every `ConsumedServiceInstance` has its own calculated random timer based on the min and max value of the `InitialSdDelayConfig`. Thus, if the calculated random timer expires, then a `FindService` entry of the referencing `ConsumedServiceInstance` shall be sent out.]()

Note: [TPS_SYST_02263] enables a scattering of `FindServices` within the configured min and max value of the `InitialSdDelayConfig`, respectively of different `ConsumedServiceInstances`.

[TPS_SYST_02184] Repetition Wait Phase configuration for a `ConsumedServiceInstance` [The Repetition Wait Phase for a `ConsumedServiceInstance` is configured with the `initialFindBehavior` and the two attributes `initialRepetitionsMax` and `initialRepetitionsBaseDelay`.]()

If the Repetition Phase is entered, the Service Discovery waits the `initialRepetitionsBaseDelay` and sends a `FindService` entry.

If the amount of sent `FindService` entries reaches `initialRepetitionsMax` and no `OfferService` is received the Main Phase will be entered. In the Main Phase no further `FindService` entries are sent by the client.

[TPS_SYST_02185] TTL for Find Service Entries [The lifetime of a `ConsumedServiceInstance` is configurable with the `serviceFindTimeToLive` attribute of `SomeipSdClientServiceInstanceConfig`.]()

If the time that is configured by `serviceFindTimeToLive` expires the `FindService` entry shall be considered not existing.

The `ConsumedServiceInstance` aggregates a `ConsumedEventGroup` in the role `consumedEventGroup` that allows to define service instance specific configuration settings for a SOME/IP EventGroup. The EventGroup specific timing settings are configured in the `SomeipSdClientEventGroupTimingConfig` that is referenced by the `ConsumedEventGroup` in the role `sdClientTimerConfig`.

Class	SomeipSdClientEventGroupTimingConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This meta-class is used to specify configuration related to service discovery in the context of an event group on SOME/IP. Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=SomeipSdTimingConfigs			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
request ResponseDelay	RequestResponseDelay	0..1	aggr	The Service Discovery shall delay answers to unicast messages triggered by multicast messages (e.g. Subscribe Eventgroup after Offer Service).
subscribe Eventgroup RetryDelay	TimeValue	0..1	attr	This attribute defines the interval in seconds to re-trigger a subscription to a Eventgroup, if a retry to subscribe to a Eventgroup is configured (subscribeEventgroupRetryMax > 0).
subscribe Eventgroup RetryMax	PositiveInteger	0..1	attr	This attribute define the maximum counts of retries to subscribe to an Eventgroup. If the value is set to 0 no retry shall be done. If the value is set to 255 the retry shall be done as long as the Eventgroup is requested and no SubscribeEventGroupAck was received.
timeToLive	PositiveInteger	1	attr	Defines the time in seconds the subscription of this event is expected by the client. this value is sent from the client to the server in the SD-subscribeEvent message.

Table 6.175: SomeipSdClientEventGroupTimingConfig

[TPS_SYST_02187] [SomeipSdClientEventGroupTimingConfig.timeToLive](#) for subscribeEventGroup Entries [The lifetime of an event subscription is configurable with the `timeToLive` attribute of `SomeipSdClientEventGroupTimingConfig` that is aggregated by an `ConsumedEventGroup` in the role `sdClientTimerConfig`.

If the time that is configured by `timeToLive` expires the event subscription is canceled.]()

It is possible to define a retry for subscription to a `ConsumedEventGroup`. The retry is optional and used by a `ConsumedServiceInstance`. It could be used to speed up the recovery if a SOME/IP-SD message is lost (e.g. `SubscribeEventGroupAck`) and the interval between cyclic offers (`SomeipSdServerServiceInstanceConfig.offerCyclicDelay`) of the corresponding `ProvidedServiceInstance` are too large to get a fast recovery, or to speed up the subscription to a `ConsumedEventGroup` if an `ConsumedEventGroup` is requested somewhere between two cyclic offers.

The retry is configurable within `SomeipSdServerServiceInstanceConfig` by setting the `subscribeEventgroupRetryMax` value greater than 0 to define how often a retry shall be triggered and `subscribeEventgroupRetryDelay` to define the timing interval the retries shall be triggered. If `subscribeEventgroupRetryMax` is set to 255, the retry to subscribe to a `ConsumedEventGroup` shall be done as long as the `ConsumedEventGroup` is requested and no acknowledgment (`SubscribeEventGroupAck`) from the `ProvidedServiceInstance` was received.

[constr_5095] Relationship between the timing behavior of the `ConsumedEventGroup` retry and the timing behavior of an Offer message [The timing behavior for a retry to a `ConsumedEventGroup` (`subscribeEventgroupRetryMax`, `subscribeEventgroupRetryDelay`) shall not overlap to the timing behavior (`SomeipSdServerServiceInstanceConfig.offerCyclicDelay`) of the corresponding `ProvidedServiceInstance`.]()

[constr_5096] `ConsumedEventGroup` with value `subscribeEventgroupRetryMax` set to 255 [Retry to a `ConsumedEventGroup` with value `subscribeEventgroupRetryMax` set to 255 is only allowed if the `SomeipSdServerServiceInstanceConfig.offerCyclicDelay` is set 0 and `serviceOfferTimeToLive` is set to 0xffffffff of the corresponding `ProvidedServiceInstance`.]()

[TPS_SYST_02188] Clients `RequestResponseDelay` for received `ServiceOffer` entries [The Client will delay the `SubscribeEventGroup` answer to a received `ServiceOffer` message by the configured `SomeipSdClientEventGroupTimingConfig.requestResponseDelay` that is aggregated by the `ConsumedEventGroup` in the role `sdClientTimerConfig`.

The actual delay will be randomly chosen between the `maxValue` and `minValue`.]()

6.7.5.8 Group of `ConsumedServiceInstances` and `ProvidedServiceInstances`

The AUTOSAR ServiceDiscovery provides a mechanism of starting/stopping `ConsumedProvidedServiceInstanceGroups`. Therefore, several `ConsumedServiceInstances` and `ProvidedServiceInstances`, respectively, could be enclosed to a `ConsumedProvidedServiceInstanceGroup`.

Class	ConsumedProvidedServiceInstanceGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	<p>The AUTOSAR ServiceDiscovery is able to start and to stop ClientServices and Server Services, respectively, at runtime. A SdServiceGroup contains several ClientServices and Server Services, respectively.</p> <p>Tags:atp.recommendedPackage=ConsumedProvidedServiceInstanceGroups</p>			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
consumed ServiceInstance	ConsumedService Instance	*	ref	<p>This reference assigns a set of ProvidedServiceInstances to the ConsumedProvidedServiceInstanceGroup.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
providedService Instance	ProvidedService Instance	*	ref	<p>This reference assigns a set of ConsumedService Instances to the ConsumedProvidedServiceInstance Group.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>

Table 6.176: ConsumedProvidedServiceInstanceGroup

The [ConsumedProvidedServiceInstanceGroup](#) is mapped to a SdServiceGroup within the ECUC configuration. The SdServiceGroups could be accessed via dedicated APIs within the AUTOSAR ServiceDiscovery (Sd) module to start and stop the requested SdServiceGroup. For example, the SdServiceGroups could be switched according the requested partial network similar to ComISignalIPduGroup and therefore the [PncMapping](#) refers the [ConsumedProvidedServiceInstanceGroup](#) in the role [pncConsumedProvidedServiceInstanceGroup](#).

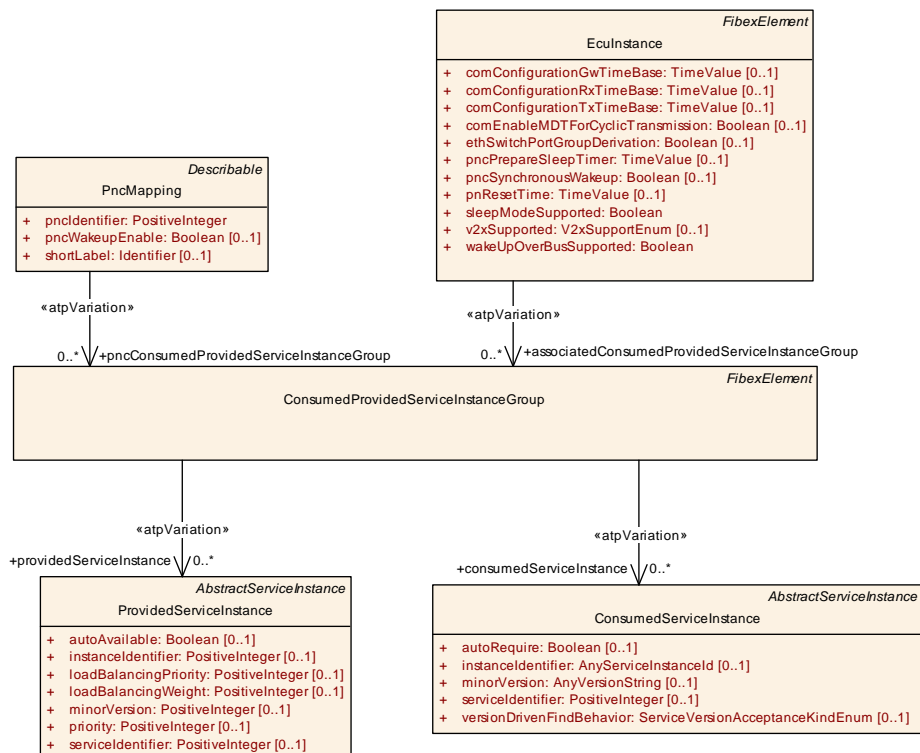


Figure 6.46: ConsumedProvidedServiceInstanceGroup

Starting a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ConsumedServiceInstances` to sent out `FindService` entries on the network and stopping a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ConsumedServiceInstances` to unsubscribe from the subscribed Event Groups and sent out an unsubscribe entry on the network.

Starting a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ProvidedServiceInstances` to sent out `OfferService` entries on the network and stopping a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ProvidedServiceInstances` to sent out a `StopOffer` entries on the network.

6.7.5.9 StaticSocketConnection

Only the SOME/IP data communication is described with `ProvidedServiceInstances` and `ConsumedServiceInstances` as described in chapter 6.7.5.3. But there is also the need to describe non-SOME/IP data exchange between two `SocketAddresses`.

[TPS_SYST_02251] Non-SOME/IP data exchange between two communication endpoints [Non-SOME/IP data exchange between two `SocketAddresses` is described with the `StaticSocketConnection` element that is aggregated by a `SocketAddress` in the role `StaticSocketConnection`. The aggregating `SocketAddress` describes one communication endpoint. The referenced `StaticSocketConnection.remoteAddress` defines the second communication endpoint.] ()

Class	StaticSocketConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Definition of static SocketConnection between the Socket that is defined by the aggregating Socket Address and the remoteAddress.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
iPduIdentifier	<code>SoConIPduIdentifier</code>	*	ref	Assignment of IPduIdentifiers that are transmitted over the static SocketConnection. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
remoteAddress	<code>SocketAddress</code>	0..1	ref	RemoteAddress of the static SocketConnection. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
tcpRole	<code>TcpRoleEnum</code>	0..1	attr	Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.

Table 6.177: StaticSocketConnection

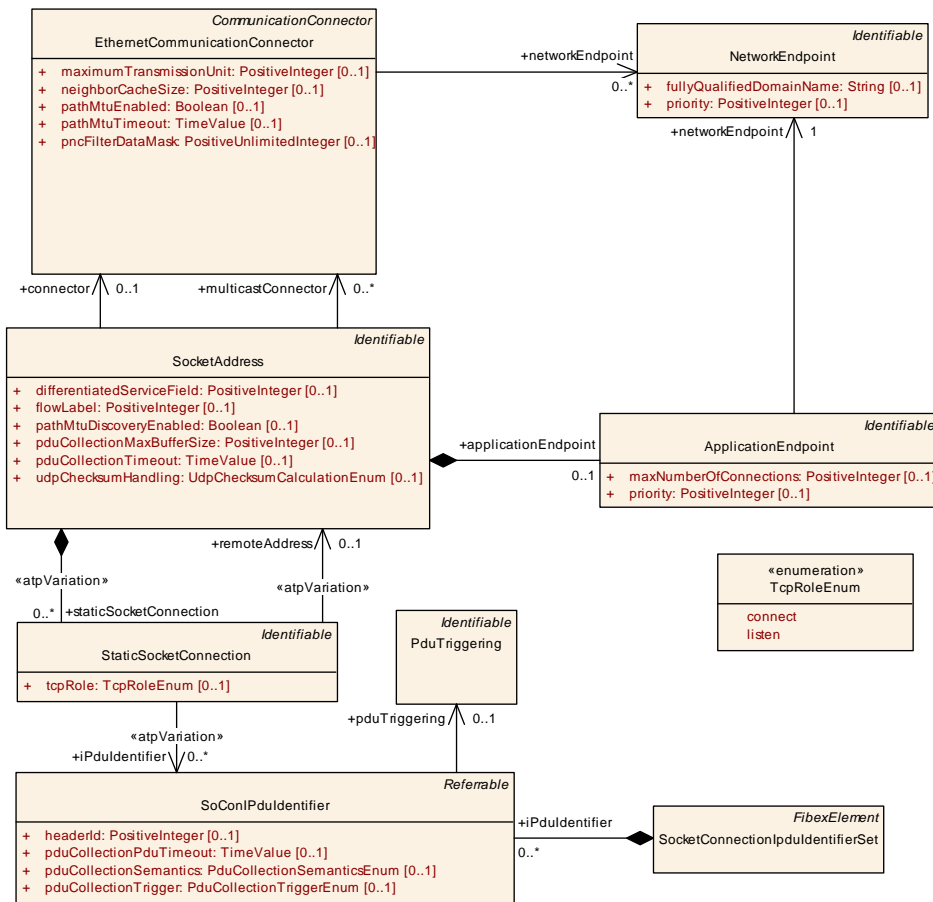


Figure 6.47: Model of StaticSocketConnection

UDP doesn't establish a connection before sending data, it is Connectionless. TCP on the other hand establishes a TCP connection. The client initiates the TCP communication and therefore it needs to be configured which endpoint acts as client and which acts as server.

[TPS_SYST_02252] Description of a TCP Client [The `SocketAddress` that aggregates the `StaticSocketConnection` with `tcpRole` set to `connect` defines the TCP client.]()

[TPS_SYST_02253] Description of a TCP Server [The `SocketAddress` that aggregates the `StaticSocketConnection` with `tcpRole` set to `listen` defines the TCP server.]()

[constr_5091] Relevance of `tcpRole` attribute [The attribute `tcpRole` is only relevant if the `StaticSocketConnection` is aggregated by a `SocketAddress` that defines a TCP Port in the aggregated `ApplicationEndpoint`.]()

[constr_5092] Relevance of `tcpRole` attribute [The transport protocol that is defined by the `SocketAddress` that aggregates the `StaticSocketConnection` shall be the same in the `SocketAddress` that is referenced by the same `StaticSocketConnection` in the role `remoteAddress`.]()

[TPS_SYST_02254] Pdus transported over the `StaticSocketConnection` [A Pdu that is transported over the `StaticSocketConnection` is described by `SoConIPduIdentifier` that is referenced in the role `iPduIdentifier`.]()

Please note that the `remoteAddress` is allowed to be defined as ANY (TpPort = 0 and ipv4address/ipv6address = ANY). In other words the remote port and/or remote IP address is configured at runtime and the ECU is able to receive all packets that are addressed to its local address no matter what the remote address is.

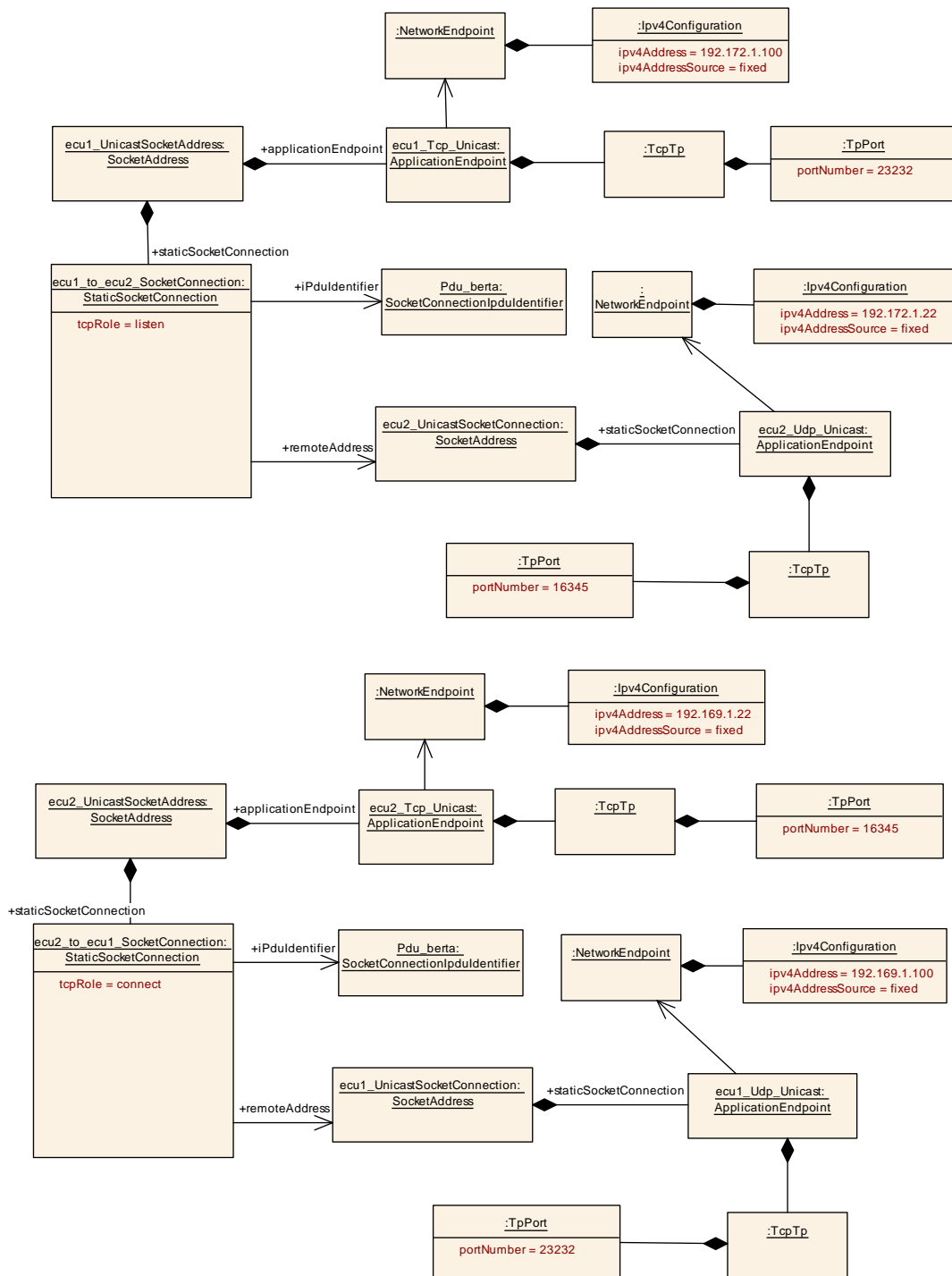


Figure 6.48: Example for the modeling of TCP StaticSocketConnections in TCP server role and TCP client role

The example in Figure 6.48 shows a `StaticSocketConnection` with the name `ecu2_to_ecu1_SocketConnection` on `ecu1_UnicastSocketAddress` that defines the TCP server. The socket connection is established to to `ecu2_UnicastSocketAddress` `remoteAddress`.

It also shows the `StaticSocketConnection` with the name `ecu1_to_ecu2_SocketConnection` on `ecu2_UnicastSocketAddress` that defines the TCP client. The socket connection is established to the `ecu1_UnicastSocketAddress` `remoteAddress`.

6.7.5.10 Service Discovery Message Configuration

If Service Discovery is used a Service Discovery Instance is configurable on an `EcuInstance` for a certain VLAN using the respective `ApplicationEndpoint`. The Service Discovery Instance refers to the configuration of a Service Discovery for a VLAN.

[TPS_SYST_02116] Modeling of Service Discovery Pdus [A Service Discovery Instance configuration requires:

- one Tx `Pdu` that is modeled as `GeneralPurposePdu` with category = SD
- one Rx `Pdu` (unicast reception) that is modeled as `GeneralPurposePdu` with category = SD
- one Rx `Pdu` (multicast reception) that is modeled as `GeneralPurposePdu` with category = SD

]()

[TPS_SYST_02117] Length of `GeneralPurposePdu` with category SD [The `length` attribute for `GeneralPurposePdus` with category = SD shall be set to at most `EthernetCommunicationConnector.maximumTransmissionUnit` - 36 Byte.]()

[TPS_SYST_02118] Rules for the creation of references to `IPduPorts` from `PduTriggerings` related to `GeneralPurposePdus` with category SD [For each `GeneralPurposePdu` with category SD a `PduTriggering` needs to be defined on the `EthernetPhysicalChannel` (VLAN) that is referenced from the `CommunicationConnector` of the `EcuInstance` on which the Service Discovery Instance is configured:

- the `PduTriggering` for the Tx `GeneralPurposePdu` references the OUT `IPduPort` of the `EcuInstance`
- the `PduTriggering` for the Rx `GeneralPurposePdu` (unicast reception) references the IN `IPduPort` of the `EcuInstance`
- the `PduTriggering` for the Rx `GeneralPurposePdu` (multicast reception) references the IN `IPduPort` of the `EcuInstance`

]()

[TPS_SYST_02119] `StaticSocketConnections` for `GeneralPurposePdus` with category SD [UDP Sockets are used for the transmission of SD messages and the following `StaticSocketConnections` shall be created:

- `StaticSocketConnection` A for all Tx and unicast Rx `GeneralPurposePdu`s
- `StaticSocketConnection` B for multicast Rx `GeneralPurposePdu`

The `PduTriggering` for the Tx `GeneralPurposePdu` and the `PduTriggering` for the Rx `GeneralPurposePdu` (unicast reception) are assigned to the `StaticSocketConnection` A via `SoConIPduIdentifier`. The `PduTriggering` for Rx Multicast `GeneralPurposePdu` is assigned to the `StaticSocketConnection` B via `SoConIPduIdentifier`.]()

[constr_3267] PduTriggerings in Service Discovery `StaticSocketConnections` [SD `StaticSocketConnections` defined in [TPS_SYST_02119] shall only refer to `PduTriggerings` which point to `GeneralPurposePdu`s of category SD.]()

[constr_3268] Service Discovery `StaticSocketConnection` aggregation by an `ApplicationEndpoint` [Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall be aggregated by an `ApplicationEndpoint` that defines a `Udp Port`.]()

[constr_3269] Service Discovery `StaticSocketConnection` `remoteAddress` reference to a `TpPort` [Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` with `Udp Port` `portNumber` set to 0. This means that the port number is dynamically assigned at runtime.]()

[constr_3270] Service Discovery `SocketConnection` `remoteAddress` reference to an IP Address [Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Address ANY (IPv4 or IPv6).]()

[constr_3272] `SoConIPduIdentifier.headerId` setting for SD `StaticSocketConnections` [The `SoConIPduIdentifier.headerId` of SD `StaticSocketConnections` defined in [TPS_SYST_02119] shall always be set to 0xFFFF8100 for SD messages.]()

[constr_3273] Service Discovery multicast `StaticSocketConnection`'s aggregation by an `ApplicationEndpoint` [The SD `StaticSocketConnection` for multicast defined in [TPS_SYST_02119] shall be aggregated by an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Multicast Address.]()

[constr_3274] Service Discovery unicast `StaticSocketConnection`'s aggregation by an `ApplicationEndpoint` [The SD `StaticSocketConnection` for unicast defined in [TPS_SYST_02119] shall be aggregated by an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Unicast Address.]()

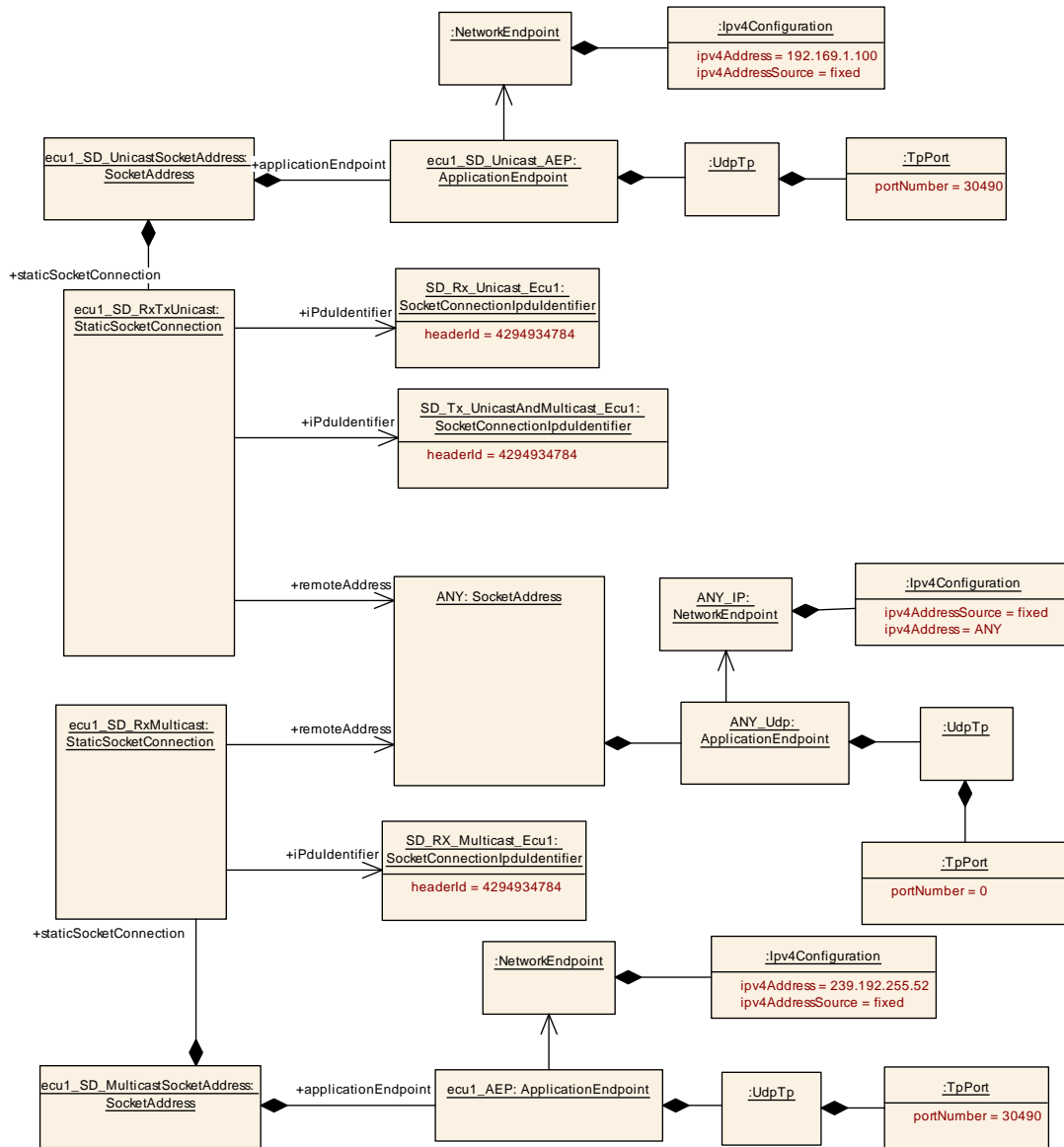


Figure 6.49: Example for the modeling of SD StaticSocketConnections

6.7.5.11 NmPdu data exchange over StaticSocketConnection

The data exchange of NmPdu on an Ethernet channel is described with the `StaticSocketConnection` as defined in chapter 6.7.5.9.

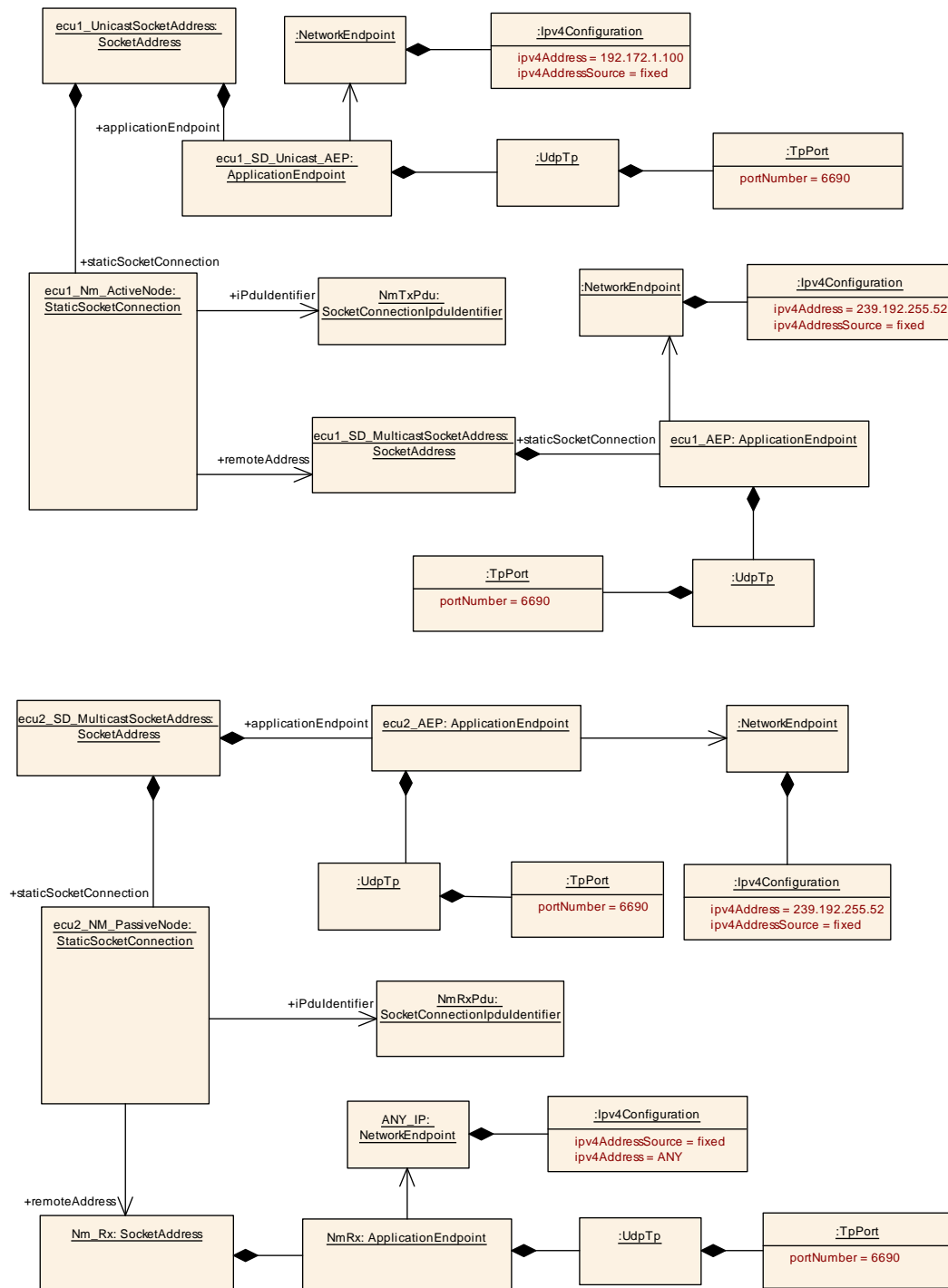


Figure 6.50: Example for the modeling of NM StaticSocketConnections

The example in [Figure 6.50](#) shows a possible configuration where the active NmNode sends the NmPdu to a remote multicast address. Each passive and active NmNode receives the NmPdu over a local multicast address from all other nodes (remote Address = ANY).

6.7.5.12 Diagnostics over IP

[DoIpConfig](#) defines a DoIP module configuration for a specific [EcuInstance](#). DoIP supports the communication of internal testers and external testers with an ECU/DoIP Node. Each DoIP node might define several logical [DoIpInterfaces](#) that may share the same physical Ethernet interface/MAC Address. The DoIP node is able to communicate on each of its [DoIpInterfaces](#) independently. I.e. the DoIP functionalities on each [DoIpInterface](#) are isolated from each other.

More details about [DoIpInterfaces](#) can be found in AUTOSAR_SWS_DiagnosticOverIP [24].

Class	DoIpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::DoIP			
Note	This element defines the DoIp configuration for a specific Ecu. Tags: atp.Status=draft			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
doipInterface	DoIpInterface	*	aggr	DoIP node consists of one or several DoIpInterfaces over which the ECU is able to communicate via DoIP independently. I.e. DoIP functionalities on each IP interface are isolated from each other. Tags: atp.Status=draft
logicAddress	DoIpLogicAddress	0..1	aggr	Describes the logical address of the DoIP entity, i.e. the Local Address that will route diagnostic requests to the Dcm of the DoIP entity. Tags: atp.Status=draft

Table 6.178: DoIpConfig

Class	DoIpInterface			
Package	M2::AUTOSARTemplates::SystemTemplate::DoIP			
Note	A logical interface over which the DoIP Node is able to communicate via DoIP independently from other existing DoIpInterfaces. Tags: atp.Status=draft			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
aliveCheckResponseTimeout	TimeValue	0..1	attr	This attribute defines the timeout in seconds for waiting for response to an Alive Check request before the connection is considered to be disconnected. Represents parameter T_TCP_AliveCheck of ISO 13400-2:2012.
doipChannelCollection	DoIpTpConfig	0..1	ref	Configuration of DoIpChannels available in an DoIp Interface. Each DoIpChannel describes a connection between a doIpSourceAddress and a doIpTargetAddress and the exchange of DcmIPdus between the PduR and DoIP. A DoIP channel is constituted by the set of all DoIp ▽





Class	DoIpInterface			
				TpConnection elements via which the configured Ecu Instance sends or receives SDUs that are sharing the same local diagnosis address and tester address. Tags:atp.Status=draft
doipConnection	SocketConnection Bundle	*	ref	DoIP Connections in the DoIpInterface that define the Do IP Pdus that are sent and received via SoAd over TCP or UDP. Tags:atp.Status=obsolete
generalInactivity Time	TimeValue	0..1	attr	This attribute defines the timeout in seconds for maximum inactivity of a TCP socket connection before the DoIP module will close the according socket connection. Represents parameter T_TCP_General_Inactivity of ISO 13400-2:2012
initialInactivity Time	TimeValue	0..1	attr	This attribute defines the timeout in seconds used for initial inactivity of a connected TCP socket connection directly after socket connection. Represents parameter T_TCP_Initial_Inactivity of ISO 13400-2:2012
initialVehicle Announcement Time	TimeValue	0..1	attr	This attribute defines the waiting time in seconds for sending first vehicle announcement message after IP address assignment. Represents parameter A_DoIP_Announce_Wait of ISO 13400-2:2012
isActivationLine Dependent	Boolean	1	attr	This attribute defines whether the network interface <ul style="list-style-type: none"> is started "on-demand" when an activation line is sensed or is always available.
maxTester Connections	PositiveInteger	0..1	attr	Maximum amount of tester connections that shall be maintained at one time before alive check is performed.
socket Connection	StaticSocketConnection	*	ref	DoIP Connections in the DoIpInterface that define the Do IP Pdus that are sent and received via SoAd over TCP or UDP. Tags:atp.Status=draft
useMacAddress ForIdentification	Boolean	0..1	attr	This attribute defines whether a configured EID at vehicle identification response/vehicle announcement is used or the MAC address. TRUE: Use MAC Address instead of EID for Vehicle identification/announcement. FALSE: Use configured EID for vehicle identification/announcement.
useVehicle Identification SyncStatus	Boolean	0..1	attr	This attribute defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.
vehicle Announcement Count	PositiveInteger	0..1	attr	This attribute defines the number of vehicle announcement messages on IP address assignment. Represents parameter A_DoIP_Announce_Num of ISO 13400-2:2012.
vehicle Announcement Interval	TimeValue	0..1	attr	This attribute defines the waiting time in seconds for sending subsequent vehicle announcement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012

Table 6.179: DoIpInterface

The [DoIpInterface](#) defines [DoIpChannels](#) with the [doipChannelCollection](#) reference that points to a [DoIpTpConfig](#) element that collects a number of [DoIpTpConnections](#).

[DoIpTpConnection](#) describes a unidirectional connection between a [doIpSourceAddress](#) and a [doIpTargetAddress](#) and the exchange of a [DcmIPdu](#) that

is defined with the `tpSdu` between the PduR and DoIP. The `DiagnosticConnection` with references to the `DoIpTpConnection` defines the related request and response messages.

A `DoIpChannel` in the Ecu configuration is constituted by the set of all `DoIpTpConnection` elements via which the `EcuInstance` that aggregates the `DoIpConfig` that references the `DoIpTpConnection` via the `DoIpInterface` sends or receives `tpSdus` that share the same local diagnosis address and tester address.

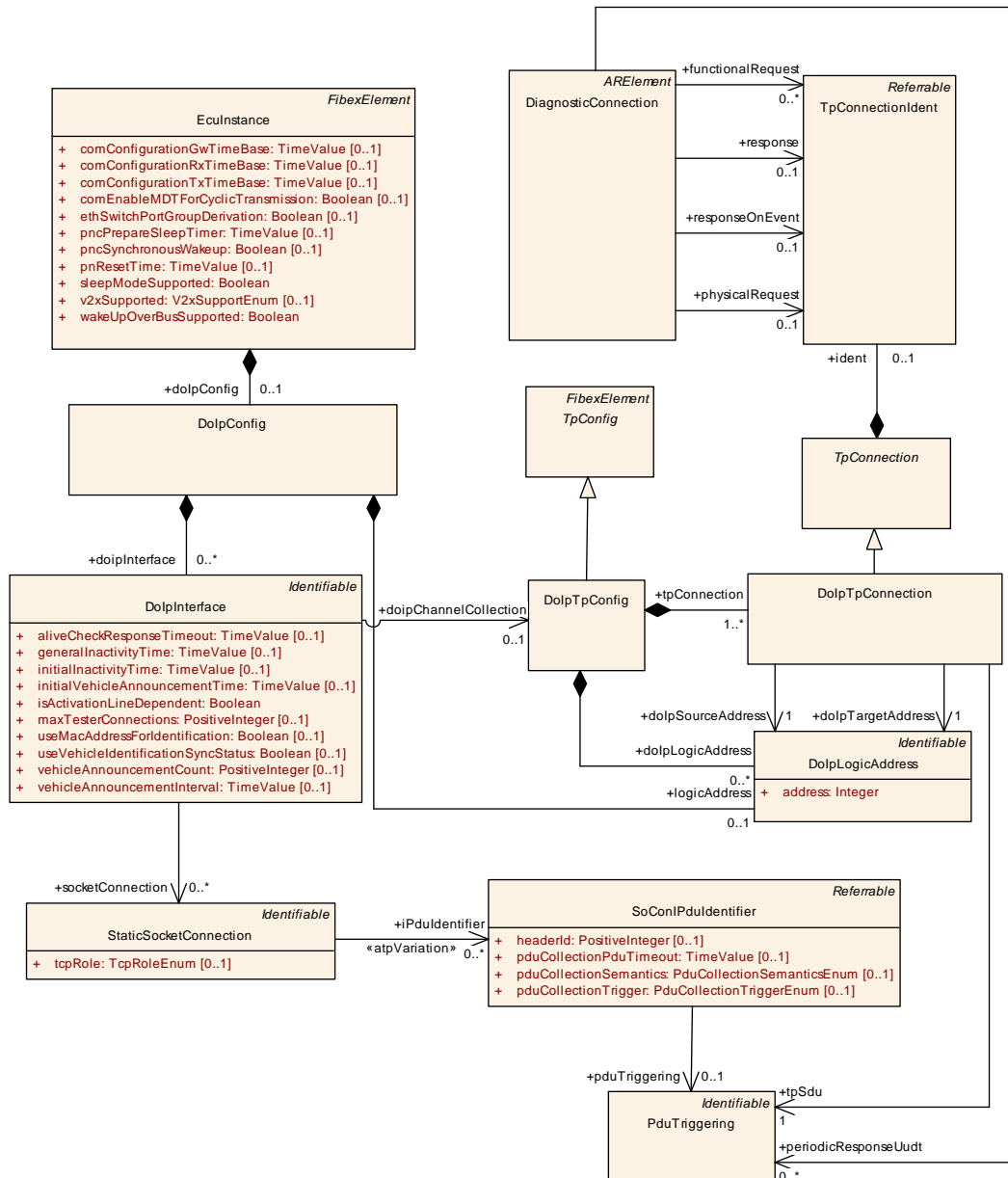


Figure 6.51: DoIP

Class	DolpTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one DolpTp Configuration that is used to configure all DolpChannels available in a DolpInterface. Each DolpChannel describes a connection between a dolpSourceAddress and a dolpTargetAddress and the exchange of DcmIPdus between the PduR and DoIP. Tags: atp.recommendedPackage=TpConfigs			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable, TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
dolpLogicAddress	DolpLogicAddress	*	aggr	Collection of logical DoIP Addresses.
tpConnection	DolpTpConnection	1..*	aggr	Collection of unidirectional connections between a source address and a target address.

Table 6.180: DolpTpConfig

Class	DolpTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	A connection identifies the sender and the receiver of this particular communication. The Dolp module routes a tpSdu through this connection.			
Base	<i>ARObject, TpConnection</i>			
Attribute	Type	Mult.	Kind	Note
dolpSourceAddress	DolpLogicAddress	1	ref	Reference to the address of the sender of the tpSdu.
dolpTargetAddress	DolpLogicAddress	1	ref	Reference to the address of the receiver of the tpSdu.
tpSdu	PduTriggering	1	ref	This reference is used to describe the data exchange between Dolp and the PduR. Only PduTriggerings of DcmIPdus shall be referenced here.

Table 6.181: DolpTpConnection

Class	DolpLogicAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	The logical DoIP address.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
address	Integer	1	attr	The logical DoIP address.

Table 6.182: DolpLogicAddress

[constr_3212] Limitation of DolpTpConnection.tpSdu [[DoIpTpConnection](#) shall only reference [PduTriggerings](#) of [DcmIPdus](#) in the role [tpSdu](#).]()

The diagnostic data is routed from the DoIP module to SoAd and back. The communication of diagnostic data via TCP/UDP is described with [StaticSocketConnections](#) that contain [SoConIPduIdentifiers](#) with references to [PduTriggerings](#) of [GeneralPurposePdus](#) of category DoIP.

Please note that there is no connection between `GeneralPurposePdus` of category DoIP and the `DoIpTpConnection` in the System Description. The DoIP module evaluates the header of an incoming `GeneralPurposePdu` and knows from the included information the further processing.

The `DoIpInterface.socketConnection` defines which `StaticSocketConnection` belongs to which `DoIpInterface`.

6.7.5.13 Transport Layer Security

AUTOSAR supports the configuration of *Transport Layer Security* for the information exchange between two *sockets*² that are modeled as `ApplicationEndpoints`.

Please note that currently the DTLS (**Datagram Transport Layer Security**) variant is not supported on the *AUTOSAR classic platform*.

It is a common use case that only one end of a TLS-based connection is actually modeled in an AUTOSAR model. The other end may exist off-board, e.g. as a diagnostic tester.

It is therefore important that the modeling does not rely on or imply knowledge about both ends of such a TLS-based connection.

An AUTOSAR model that only describes one end of the communication is positively required to work, independently of the availability of a formal modeling of the other end.

²TLS connections are - by design - limited to a 1:1 pattern. A 1:n or n:1 communication pattern is not supported by TLS

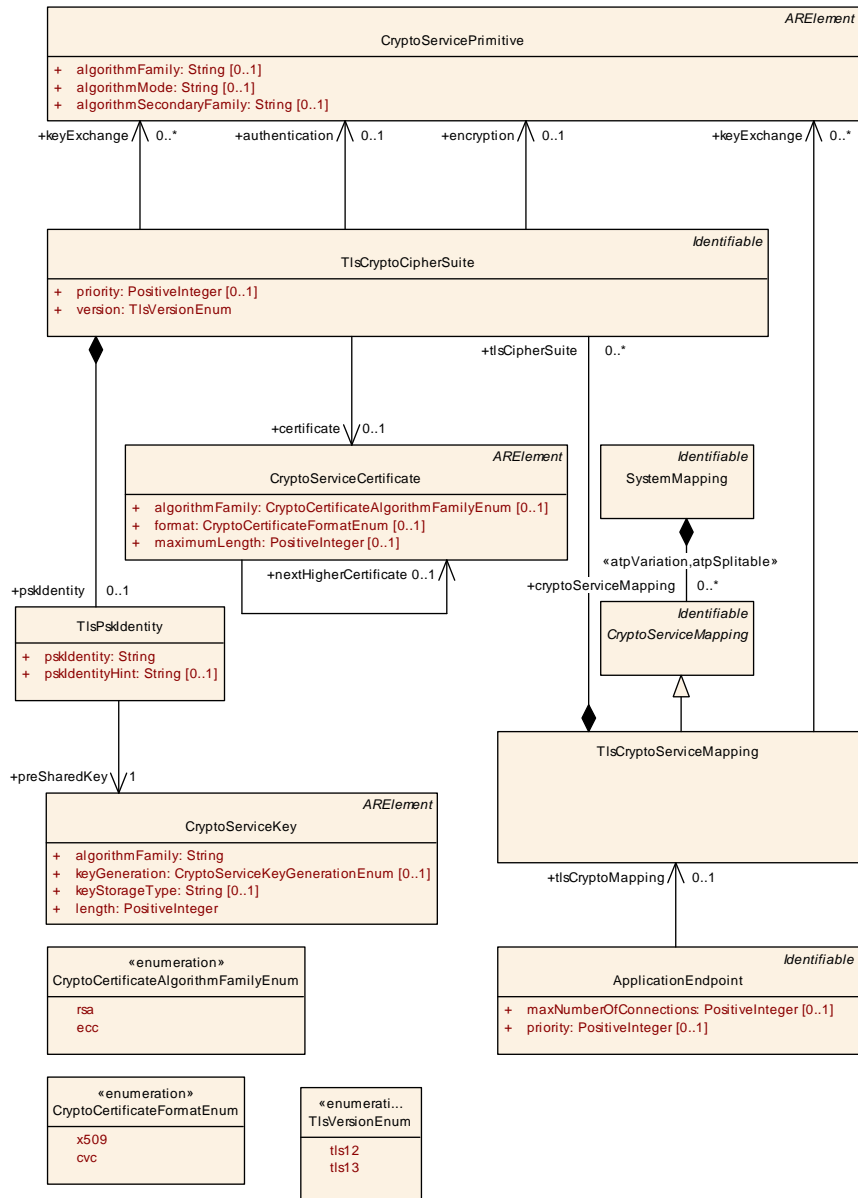


Figure 6.52: Modeling of crypto infrastructure for Transport Layer Security

Class	TlsCryptoServiceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS). Tags: atp.Status=draft atp.recommendedPackage=CryptoServiceMappings			
Base	ARObject, CryptoServiceMapping, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note





Class	TlsCryptoServiceMapping			
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. Tags: atp.Status=draft
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	This aggregation represents the collection of supported cipher suites. Tags: atp.Status=draft

Table 6.183: TlsCryptoServiceMapping

A TLS connection is established between two communication endpoints that assume the dedicated roles of *server* and *client*.

These roles cannot be swapped while the connection exists, i.e. a *server* remains the *server* for the full amount of time the connection exists.

[TPS_SYST_05029] Semantics of meta-class [TlsCryptoServiceMapping](#) [As a sub-class of [CryptoServiceMapping](#), meta-class [TlsCryptoServiceMapping](#) has the ability to collect the TLS-related configuration aspects from either the perspective of the client or the server.

In the case of TLS, the collection boils down to the aggregation of meta-class [TlsCryptoCipherSuite](#) in the role [tlsCipherSuite](#) plus the ability (by means of the role [keyExchange](#)) to define handshake properties that are shared for each of the aggregated [tlsCipherSuites](#).]()

[constr_1670] Prohibition of usage of [tlsCryptoMapping](#) in case of UDP socket connections [A [TlsCryptoServiceMapping](#) may only be referenced by an [ApplicationEndpoint](#) in the role [tlsCryptoMapping](#) if that [ApplicationEndpoint](#) aggregates a [TcpTp](#) in the role [tpConfiguration](#).]()

[constr_1671] Supported values of [TlsCryptoServiceMapping.category](#) [The only supported values of attribute [TlsCryptoServiceMapping.category](#) are:

- **TLS_SERVER:** the [TlsCryptoServiceMapping](#) assumes the role of the *server* in the TLS connection.
- **TLS_CLIENT:** the [TlsCryptoServiceMapping](#) assumes the role of the *client* in the TLS connection.

]()

The usage of a cipher suite in the context of setting up a TLS connection is formalized by means of meta-class [TlsCryptoCipherSuite](#).

[TPS_SYST_05030] Semantics of [TlsCryptoCipherSuite](#) [The creation of a TLS connection requires the usage of a suite of cryptographic operations in specific roles, also known as a *cipher suite*.

Meta-class [TlsCryptoCipherSuite](#) represents a given cipher suite for a TLS connection. [TlsCryptoCipherSuite](#) references meta-class [CryptoServicePrimitive](#) in three dedicated roles that represent the steps of the creation of a TLS connection.

More specifically, the cryptographic operations for setting up a TLS connection involve the following steps:

- **Key exchange:** these [CryptoServicePrimitives](#) may be used for the handshake phase of the TLS connection. Different alternatives exist for executing this phase and therefore the multiplicity of this reference is 0..*.
- **Authentication** of communication partners during the operational phase of the TLS connection. This part is similar to freshness calculation for SecOC-based communication. For this purpose a single [CryptoServicePrimitive](#) is used on each end of the communication.
- **Encryption** of content exchanged between the communication partners that have established the TLS connection. For this purpose a single [CryptoServicePrimitive](#) is used on each end of the communication.

]()

Class	TlsCryptoCipherSuite			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents a cipher suite for describing cryptographic operations in the context of establishing a connection of ApplicationEndpoints that is protected by TLS. Tags: atp.ManifestKind=ServiceInstanceManifest atp.Status=draft			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
authentication	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the generation and verification of MACs. Tags: atp.Status=draft
certificate	CryptoServiceCertificate	0..1	ref	This reference identifies the applicable certificate. Tags: atp.Status=draft
encryption	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the execution of encryption. Tags: atp.Status=draft
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. Tags: atp.Status=draft
priority	PositiveInteger	0..1	attr	This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.





Class	TlsCryptoCipherSuite			
pskIdentity	TlsPskIdentity	0..1	aggr	Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key. Tags: atp.Status=draft
version	TlsVersionEnum	1	attr	This attribute supports the definition of the applicable version of TLS.

Table 6.184: TlsCryptoCipherSuite

Enumeration	TlsVersionEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	This meta-class has the ability to identify a specific version of the transport-layer security (TLS) protocol. Tags: atp.Status=draft
Literal	Description
tls12	TLS version 1.2 Tags: atp.EnumerationLiteralIndex=0
tls13	TLS version 1.3 Tags: atp.EnumerationLiteralIndex=2

Table 6.185: TlsVersionEnum

Class	TlsPskIdentity			
Package	M2::AUTOSARTemplates::AdaptivePlatform::ServiceInstanceManifest::SecureCommunication			
Note	This element is used to describe the pre-shared key shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key. Tags: atp.Status=draft			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
preSharedKey	CryptoServiceKey	1	ref	This reference identifies the applicable cryptographic key. Tags: atp.Status=draft
pskIdentity	String	1	attr	This attribute provides the key identification.
pskIdentityHint	String	0..1	attr	This attribute provides the identity hint for a pre-shared key.

Table 6.186: TlsPskIdentity

[TPS_SYST_05031] Existence of [TlsCryptoCipherSuite.keyExchange](#) vs. [TlsCryptoServiceMapping.keyExchange](#) [The role [TlsCryptoServiceMapping.keyExchange](#) has been introduced as an optimization.

It is assumed that the references for key exchange look pretty similar if not identical for many concrete [TlsCryptoCipherSuites](#).

Adding these references in an identical form to a bunch of `TlsCryptoCipherSuites` does not really make sense. Therefore, `TlsCryptoServiceMapping` allows to define these references as well with the intention to make them valid for all `TlsCryptoServiceMapping.tlsCipherSuite`.

A mixture of references in the role `TlsCryptoCipherSuite.keyExchange` and `TlsCryptoServiceMapping.keyExchange` is supported for the case that a given collection of `TlsCryptoCipherSuites`]()

[TPS_SYST_05032] Semantics of `CryptoServiceCertificate` [Meta-class `CryptoServiceCertificate` represents a cryptographic certificate needed for the creation of a TLS connection between *server* and *client*.]()

Class	CryptoServiceCertificate			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents the ability to model a cryptographic certificate. Tags: atp.ManifestKind=ServiceInstanceManifest atp.Status=draft atp.recommendedPackage=CryptoServiceCertificates			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
algorithmFamily	CryptoCertificateAlgorithmFamilyEnum	0..1	attr	This attribute represents a description of the family of crypto algorithm used to generate public key and signature of the cryptographic certificate.
format	CryptoCertificateFormatEnum	0..1	attr	This attribute can be used to provide information about the format used to create the certificate
maximumLength	PositiveInteger	0..1	attr	This attribute represents the ability to define the maximum length of the certificate.
nextHigherCertificate	CryptoServiceCertificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain. Tags: atp.Status=draft

Table 6.187: CryptoServiceCertificate

Enumeration	CryptoCertificateAlgorithmFamilyEnum		
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication		
Note	This meta-class defines possible cryptographic algorithm families used to create public keys and signatures within the certificate. Tags: atp.Status=draft		
Literal	Description		
ecc	The cryptographic operations in the certificate are executed using elliptic curves (ecc) Tags: atp.EnumerationLiteralIndex=2		
rsa	The cryptographic operations in the certificate are executed using the RSA approach. Tags: atp.EnumerationLiteralIndex=1		

Table 6.188: CryptoCertificateAlgorithmFamilyEnum

Enumeration	CryptoCertificateFormatEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	This meta-class defines possible formats of cryptographic certificates. Tags: atp.Status=draft
Literal	Description
cvc	The certificate has been created in Card Verifiable Certificate (CVC) format Tags: atp.EnumerationLiteralIndex=2
x509	The certificate is created in X.509 format. Tags: atp.EnumerationLiteralIndex=1

Table 6.189: CryptoCertificateFormatEnum

[constr_1672] Existence of `TlsCryptoCipherSuite.certificate` and `TlsCryptoCipherSuite.pskIdentity` in the server role [Either

- the reference to `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite.certificate`
- the aggregation of `TlsPskIdentity` in the role `TlsCryptoCipherSuite.pskIdentity`

shall exist if the `TlsCryptoCipherSuite` is aggregated by a `TlsCryptoServiceMapping` that has attribute `category` set to the value `TLS_SERVER`.]()

In other words two different approaches are supported by TLS for the handling of key compromise: Pre-shared secret and certificate.

The server may optionally request a certificate from the *client*. If this option is not used then other documented approaches for completing the handshake phase is foreseen for the specific case.

[TPS_SYST_05033] Existence of `TlsCryptoCipherSuite.certificate` and `TlsCryptoCipherSuite.pskIdentity` in the client role [The client (`TlsCryptoServiceMapping` has the attribute `category` set to the value `TLS_CLIENT`) has the following authentication options:

- the reference to `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite.certificate` exists,
- the aggregation of `TlsPskIdentity` in the role `TlsCryptoCipherSuite.pskIdentity` exists,
- neither one or the other exists. In this case the handshake is provided on the basis of the server certificate only.

]()

In the pre-shared Key approach the client indicates which key to use by including a `pskIdentity` in the ClientKeyExchange message. To help the client in selecting

which identity to use, the server can provide a `pskIdentityHint` in the `ServerKeyExchange` message. Please note that the usage of `pskIdentityHints` is restricted for usage with TLS 1.2.

6.7.5.14 IPsec

IPsec is a protocol suite that provides cryptographic protection for IP datagrams in IPv4 and IPv6 network packets.

IPsec uses a security association to specify security properties that are shared between the communicating parties. The security association defines a relationship between two or more parties and determines which security services will be used to communicate securely. In other words the security association serves as a “contract” between the different devices.

A single security association protects data in one communication direction. Two security associations shall be present to secure traffic in both directions. Each security association can provide encryption, data integrity and data authentication.

In addition the senders and receivers of IP datagrams can determine the required protection for an IP packet according to IPsec security policies.

These are rules that define how datagrams are processed that are received by a device. For example, security policies are used to decide if a particular packet needs to be dropped or needs to be processed by IPsec.

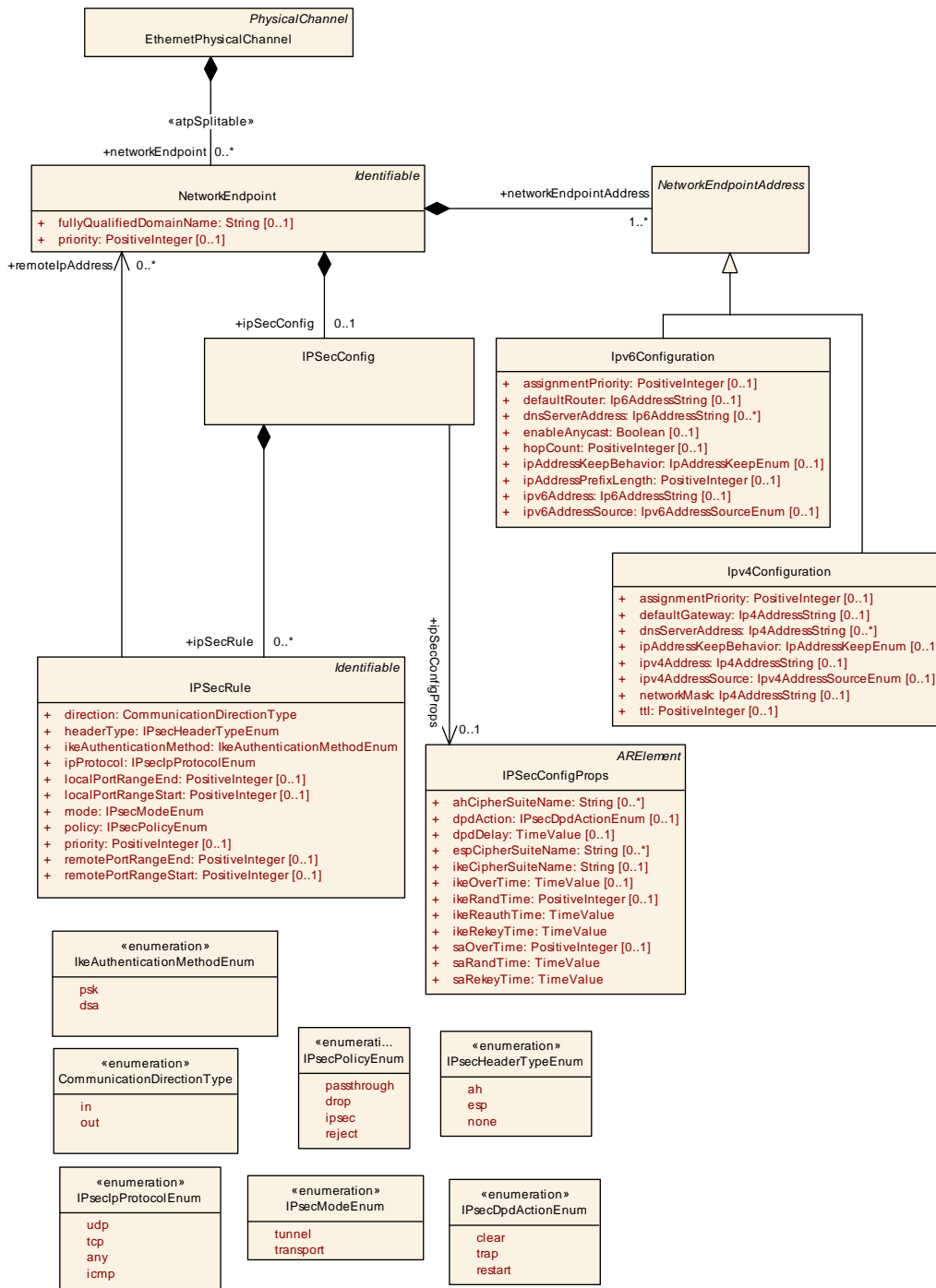


Figure 6.53: IPsec configuration model

[TPS_SYST_02265] Configuration of IPsec [The `IPsecConfig` meta-class that is aggregated by a `NetworkEndpoint` in the role `ipSecConfig` provides the ability to define IPsec settings that are necessary to configure IPsec security associations and IPsec security policies.] ()

[TPS_SYST_02266] Definition of IPsecRules [The `IPsecConfig` meta-class may contain one or several `IPsecRules`. Each `IPsecRule` defines the network connection that is monitored by IPsec by defining the local endpoint and the remote endpoint.

Each endpoint is defined by the IP Address and the Tcp/Udp Port. The communication direction for which the `IPSecRule` is valid is defined by the `direction` attribute.]()

Class	IPSecConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	IPsec is a protocol that is designed to provide "end-to-end" cryptographically-based security for IP network connections.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
ipSecConfig Props	IPSecConfigProps	0..1	ref	Global IPsec configuration settings that are valid for all IPSecRules that are defined on the NetworkEndpoint.
ipSecRule	IPSecRule	*	aggr	IPSec rules and filters that are defined in the IPSecConfig for a specific NetworkEndpoint.

Table 6.190: IPSecConfig

Class	IPSecRule			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
direction	CommunicationDirectionType	1	attr	This attribute defines the direction in which the traffic is monitored. If this attribute is not set a bidirectional traffic monitoring is assumed.
headerType	IPsecHeaderTypeEnum	1	attr	Header type specifying the IPsec security mechanism.
ike Authentication Method	IkeAuthenticationMethodEnum	1	attr	This attribute defines the IKE authentication method that is used locally and is expected on the remote side.
ipProtocol	IPsecIpProtocolEnum	1	attr	This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.
localPortRange End	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring to tcp and defines a end value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
localPortRange Start	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring to tcp and defines a start value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
mode	IPsecModeEnum	1	attr	This attribute defines the type of the connection.
policy	IPsecPolicyEnum	1	attr	An IPsec policy defines the rules that determine which type of IP traffic needs to be secured using IPsec and how that traffic is secured.





Class	IPSecRule			
priority	PositiveInteger	0..1	attr	This attribute defines the priority of the IPSecRule (SPD entry). The processing of entries is based on priority, starting with the highest priority "0".
remoteIp Address	NetworkEndpoint	*	ref	Definition of the remote NetworkEndpoint. With this reference the connection between the local Network Endpoint and the remote NetworkEndpoint is described on which the traffic is monitored.
remotePort RangeEnd	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring to tcp and defines a end value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
remotePort RangeStart	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring to tcp and defines a start value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.

Table 6.191: IPSecRule

Class	IPSecConfigProps			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This element holds all the attributes for configuration of IPsec that are independent of specific IPsec rules. Tags: atp.recommendedPackage=IPSecConfigProps			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
ahCipherSuite Name	String	*	attr	AH (Authentication Header) algorithm to be used for the connection, e.g. HMAC/SHA2-256
dpdAction	IPsecDpdActionEnum	0..1	attr	This attribute defines what to do if the peer is considered dead. If not configured "restart" shall be assumed.
dpdDelay	TimeValue	0..1	attr	This attribute describes the interval to check the liveness of a peer actively using IKEv2 INFORMATIONAL exchanges. Active DPD checking is only enforced if no IKE or ESP/AH packet has been received for the configured DPD delay. In not configured the value "5 minutes" shall be assumed.
espCipherSuite Name	String	*	attr	ESP (Encapsulating Security Payload) algorithm that provides encryption and optional authentication for the connection, e.g. AES-128+SHA2-256.
ikeCipherSuite Name	String	0..1	attr	IKE encryption/authentication algorithms to be used for the connection.





Class	IPSecConfigProps			
ikeOverTime	TimeValue	0..1	attr	This attribute describes the hard deadline when an SA becomes invalid in percentage. Example: ikeOverTime of max(ikeReauthTime, ikeRekeyTime). Default: 10 %
ikeRandTime	PositiveInteger	0..1	attr	This attribute defines in percentage by how long before the expiration of ikeReauthTime and ikeRekeyTime will be rekeyed/reauthenticated. Default: 10%
ikeReauthTime	TimeValue	1	attr	This attribute defines the absolute time after which an IKE SA will be reauthenticated. 0 means reauthentication is disabled.
ikeRekeyTime	TimeValue	1	attr	This attribute defines the absolute time after which an IKE SA will be rekeyed. 0 means rekey is disabled.
saOverTime	PositiveInteger	0..1	attr	This attribute describes the hard deadline when an IPsec SA becomes invalid in percentage. Example: saOverTime * saRekeyTime. Default: 110%
saRandTime	TimeValue	1	attr	This attribute defines by how long before the expiration of saRekeyTime will be rekeyed.
saRekeyTime	TimeValue	1	attr	This attribute defines the absolute time after which an IPsec SA will be rekeyed. 0 means rekey is disabled.

Table 6.192: IPSecConfigProps

Enumeration	IPsecIpProtocolEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	Definition of supported TcpIp protocols that are supported in Security Policy Database (SPD) entries in IPsec configurations.
Literal	Description
any	ANY protocol Tags: atp.EnumerationLiteralIndex=3
icmp	Internet Control Message Protocol (ICMP) Tags: atp.EnumerationLiteralIndex=2
tcp	TCP Protocol Tags: atp.EnumerationLiteralIndex=1
udp	UDP Protocol Tags: atp.EnumerationLiteralIndex=0

Table 6.193: IPsecIpProtocolEnum

[TPS_SYST_02270] Definition of general IPsec configuration settings [General configuration properties that are independent of particular *IPSecRules* are collected in the *IPSecConfigProps* element that is referenced from the *IPSecConfig* in the role *ipSecConfigProps*.]()

[TPS_SYST_02267] **IPsec policy** [The `IPSecRule.policy` attribute defines how IP packets are handled that are going over the network connection defined by the `IPSecRule`. In detail it defines whether the IP packet is processed by IPsec or not.]()

<i>Enumeration</i>	IPsecPolicyEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	
Literal	Description
drop	Signifying that packets should be discarded Tags: atp.EnumerationLiteralIndex=3
ipsec	Signifying that packets should be protected. Tags: atp.EnumerationLiteralIndex=1
passthrough	Signifying that no IPsec processing should be done at all. Tags: atp.EnumerationLiteralIndex=2
reject	Signifying that packets should be discarded and a diagnostic ICMP returned. Tags: atp.EnumerationLiteralIndex=4

Table 6.194: IPsecPolicyEnum

IPsec can be configured to operate in two different modes, Tunnel and Transport mode. With tunnel mode, the entire IP packet is protected by IPsec. IPsec wraps the original packet, encrypts it and adds a new IP header to it. The tunnel mode is most commonly used between VPN gateways and the IP addresses of the newly added outer IP header are that of the VPN Gateways. In other words the traffic between the two VPN Gateways is protected and each gateway acts as a proxy for the hosts behind it.

The transport mode provides the protection of the Data Payload of the IP datagram with an AH or ESP header. The IP Header remains the same and IPsec inserts its header between the IP header and the upper level headers. The IPsec transport mode can be used when securing traffic between two hosts or between a host and a VPN gateway.

[TPS_SYST_02268] **IPsec mode** [The `IPSecRule.mode` attribute defines whether the IP packet is processed in the `transport` or `tunnel` mode.]()

Please note that AUTOSAR currently supports only the `transport mode` as configuration option.

<i>Enumeration</i>	IPsecModeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	This enumeration describes the supported IPsec modes.
Literal	Description
transport	Signifying that the IPsec transport mode is used. With the transport mode the original IP header is retained and only the IP payload and ESP trailer is encrypted. Tags: atp.EnumerationLiteralIndex=1





Enumeration	IPsecModeEnum
tunnel	Signifying that the IPsec tunnel mode is used. With tunnel mode, the entire original IP packet is protected by IPsec. This means IPsec wraps the original packet, encrypts it, adds a new IP header and sends it to the other side. Tags: atp.EnumerationLiteralIndex=0

Table 6.195: IPsecModeEnum

IPsec uses two protocols:

- AH - Authentication Header
- ESP - Encapsulating Security Payload

The AH protocol provides a mechanism for authentication only and authenticates the entire IP packet, including the outer IP header.

The ESP protocol provides data confidentiality (encryption) and/or authentication (data integrity, data origin authentication, and replay protection).

When ESP is used in transport mode, the IP payload is encrypted and the original IP header is moved to the front of the message. The ESP header is inserted after the IP header and is signed together with the IP payload. The original IP header remains unprotected.

When ESP is used in tunnel mode a new IP Header is created and the ESP header is added in front of the original IP Packet. The entire original IP packet is encrypted and signed in this mode.

[TPS_SYST_02269] IPsec AH and ESP protocol configuration [In the [IPsecRule](#) it is possible to define the IPsec protocol that shall be used to protect IP packets that are going over the defined network connection. The attribute [headerType](#) defines whether AH, ESP or neither one is used.] ()

Enumeration	IPsecHeaderTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	IPsec Header Type options
Literal	Description
ah	Authentication Header (AH) Tags: atp.EnumerationLiteralIndex=0
esp	Encapsulating Security Payloads (ESP) Tags: atp.EnumerationLiteralIndex=1
none	No header Tags: atp.EnumerationLiteralIndex=2

Table 6.196: IPsecHeaderTypeEnum

[TPS_SYST_02271] IPsec AH and ESP CipherSuites [The attributes [ahCipherSuiteName](#) and [espCipherSuiteName](#) define the supported AH and ESP algorithms.] ()

The naming convention for `ahCipherSuiteName`, `espCipherSuiteName` and `IPSecConfigProps.ikeCipherSuiteName` shall follow the naming convention for cryptographic primitives that is defined in [25].

[TPS_SYST_02272] IPsec Internet Key Exchange protocol configuration [In the `IPSecRule` it is possible to define how IKE protocol authenticates other party. The usage of the attribute `ikeAuthenticationMethod` defines whether Digital Signature Authentication or pre-shared key is used.]()

<i>Enumeration</i>	IkeAuthenticationMethodEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	This enumeration describes the supported IKE authentication methods.
Literal	Description
dsa	Digital Signature Authentication Tags: atp.EnumerationLiteralIndex=2
psk	Pre-shared key authentication Tags: atp.EnumerationLiteralIndex=1

Table 6.197: IkeAuthenticationMethodEnum

Please note that the supported IKE CipherSuites are configured with the `IPSecConfigProps.ikeCipherSuiteName`. The `IPSecConfigProps` contains additional IKE specific configuration settings.

<i>Enumeration</i>	IPSecDpdActionEnum
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication
Note	Potential Dead Peer Detection (Dpd) Actions
Literal	Description
clear	Deletes the SA. Tags: atp.EnumerationLiteralIndex=0
restart	Immediately tries to establish the connection. Tags: atp.EnumerationLiteralIndex=2
trap	tries to establish the connection after traffic is sent to the peer. Tags: atp.EnumerationLiteralIndex=1

Table 6.198: IPSecDpdActionEnum

[TPS_SYST_02273] Protection of `ProvidedServiceInstance` by IPsec [To describe the protection of an `ProvidedServiceInstance` by IPsec the `ProvidedServiceInstance` needs to point to an `ApplicationEndpoint` in the `localUnicastAddress` role and this `ApplicationEndpoint` shall point to a `NetworkEndpoint` that aggregates the `IPSecConfig` and describes the IPsec Security Associations.]()

[TPS_SYST_02274] Protection of ConsumedServiceInstance by IPsec [To describe the protection of an ConsumedServiceInstance by IPsec the ConsumedServiceInstance needs to point to an ApplicationEndpoint in the localUnicastAddress role and this ApplicationEndpoint shall point to a NetworkEndpoint that aggregates the IPsecConfig and describes the IPsec Security Associations.]()

Please note that IP Multicast protection by IPsec is not supported.

6.7.5.15 EthernetFrameType based communication

Please note that with the introduction of the Tcplp Bsw module the description of AbstractEthernetFrames is no longer necessary for configuration of the AUTOSAR Tcplp Stack.

Nevertheless it may be useful to describe the Ethernet FrameType based communication in some cases, e.g. if a new basic software module like leee1722Tp is used that is located above the EthDrv and parallel to the Tcplp Stack. The Ethernet FrameType based communication shall be described without Pdus.

[constr_3113] AbstractEthernetFrame shall not have a PduToFrameMapping [It is not allowed to map Pdus into AbstractEthernetFrames.]()

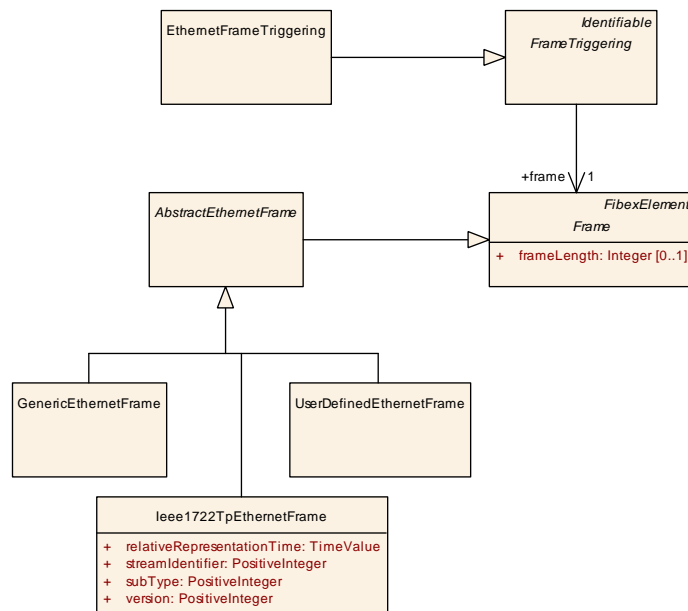


Figure 6.54: EthernetFrameType based communication

Class	AbstractEthernetFrame (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	Ethernet specific attributes to the Frame.			
Base	ARObject, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	GenericEthernetFrame, Ieee1722TpEthernetFrame, UserDefinedEthernetFrame			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.199: AbstractEthernetFrame

Class	EthernetFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	Ethernet specific Frame element.			
Base	ARObject, FrameTriggering, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.200: EthernetFrameTriggering

Class	GenericEthernetFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	This element is used for EthernetFrames without additional attributes that are routed by the EthIf. Tags: atp.recommendedPackage=Frames			
Base	ARObject, AbstractEthernetFrame, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.201: GenericEthernetFrame

Class	UserDefinedEthernetFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	UserDefinedEthernetFrame allows the description of a frame-based communication to Complex Drivers that are located above the EthDrv. Tags: atp.recommendedPackage=Frames			
Base	ARObject, AbstractEthernetFrame, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.202: UserDefinedEthernetFrame

Class	ieee1722TpEthernetFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	ieee1722Tp Ethernet Frame Tags: atp.recommendedPackage=Frames			
Base	<i>ARObject, AbstractEthernetFrame, CollectableElement, FibexElement, Frame, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
relativeRepresentationTime	TimeValue	1	attr	Defines the time when content shall be presented (in seconds). The actual absolute time is creation time plus relative presentation time
streamIdentifier	PositiveInteger	1	attr	IEEE 1722 stream identifier.
subType	PositiveInteger	1	attr	Protocol type.
version	PositiveInteger	1	attr	Revision of ieee1722 standard.

Table 6.203: ieee1722TpEthernetFrame

6.8 Transport Layer

In AUTOSAR, the Transport Layer has two main purposes: The segmentation and re-assembly of messages that are too long to fit into one frame on the underlying communication cluster, and the re-use of fixed frame identifiers for different message content.

According to the AUTOSAR Layered Software Architecture [15], each type of communication cluster has its own definition of the Transport Layer. Consequently, the peculiarities of the cluster types are addressed in the System Template by having different detailed models for FlexRay, CAN, LIN and J1939. However, all models are embedded into the communication model: They use specialized classes of `TpConfig` as a root element into the TP configuration.

[TPS_SYST_01099] Context of `TpConfig` [A `TpConfig` element is existing always in the context of exactly one `CommunicationCluster`.] (*RS_SYST_00014*)

All Transport Layers will take `IPdus` as input elements, which will be transferred in the form of one or more `NPdus`. A `TpConnection` (`FlexrayTpConnection`, `CanTpConnection`, `LinTpConnection`, `J1939TpConnection`) identifies a connection link between different communication nodes and routes the `Pdus` between them.

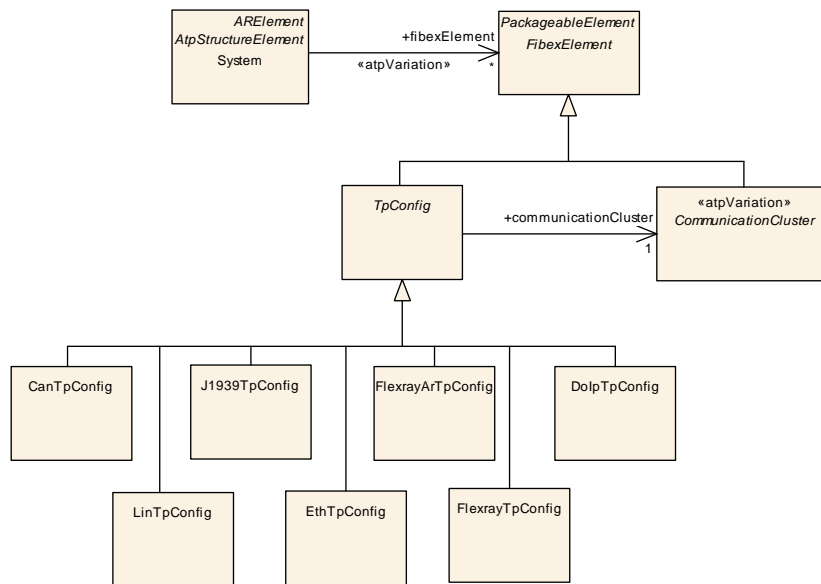


Figure 6.55: Transport Layer Overview

Examples in chapter 6.8.9 and chapter 6.8.10 illustrate the usage of the TP model.

Class	<code>TpConfig</code> (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Contains all configuration elements for AUTOSAR TP.
Base	<code>ARObject</code> , <code>CollectableElement</code> , <code>FibexElement</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PackageableElement</code> , <code>Referrable</code>





Class	<i>TpConfig</i> (abstract)			
Subclasses	CanTpConfig, DolpTpConfig, EthTpConfig, FlexrayArTpConfig, FlexrayTpConfig, J1939TpConfig, LinTpConfig, SomeipTpConfig			
Attribute	Type	Mult.	Kind	Note
communication Cluster	CommunicationCluster	1	ref	A TpConfig is existing always in the context of exactly one CommunicationCluster.

Table 6.204: TpConfig

Class	TpAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
tpAddress	Integer	1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.

Table 6.205: TpAddress

[constr_3025] Usage of NPdus in TpConnections [In case several TpConnections use the same Frame ID for their communication needs only one NPdu element per Frame Id shall exist. This constraint applies for all supported AUTOSAR transport protocols (CanTp, LinTp, FrTp, FrArTp and J1939Tp).]()

Note: Depending on the capabilities of the Basic Software implementations of Tp and Interface the ECU Configuration of the respective BSW Modules may utilize more communication elements (NPdus).

Example for an allowed System Template description where the same Frameld is used by two different TpConnections:

```

TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu2 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu1 ----> FrameId1
    
```

The following Ecu configuration with additional NPdus can still be derived from the above system description:

```

TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu3 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu4 ----> FrameId1
    
```

[constr_3090] Tpsdu transmission on a PhysicalChannel [The IPdu that is referenced by a TpConnection in the role tpSdu shall be referenced by exactly one PduTriggering aggregated on the PhysicalChannel of the TpConnection.]()

The corresponding PduTriggering for the IPdu referenced from the TpConnection in the role tpSdu is aggregated by the PhysicalChannel which points to the

same `CommunicationConnector` which is referenced by `TpNode` that this `TpConnection` points to.

Please note that with [\[constr_3090\]](#) the multiple transmission of the same `TpSdu` over a specific channel using TP is only possible if several `IPdus` and `TpConnections` are created.

6.8.1 Transport Layer Routing

The transformations in the TP modules take a significant amount of time and resources and therefore two different Transport Layer routing approaches are supported by AUTOSAR.

[TPS_SYST_01100] TP routing without using transport protocol modules (low-level routing) [The behavior can be optimized if source and target use the same transport protocol (e.g. CanTp-to-CanTp routing). In this case the inbound `NpDU` can be directly forwarded to the `PduR` and then sent on the outbound bus without any (resource consuming) TP module involvement.] [\(RS_SYST_00014\)](#)

[TPS_SYST_01101] TP routing using transport protocol modules [In case that transport protocol modules are involved in the routing operation the incoming `NpDUs` need to be:

- forwarded to corresponding inbound TP module and reassembled into an `SDU` (represented as `IPdu`)
- the `SDU` needs to be forwarded to the `PduR`
- the `PduR` routes the `SDU` to the outgoing TP module
- the outbound TP module segments the `SDU` into `NpDUs` which are then sent on the target bus.

] [\(RS_SYST_00014\)](#)

6.8.2 FlexRay ISO Transport Layer

The FlexRay ISO 10681-2 Transport Layer supports multiple sessions, i.e. multiple segmented transfers can be handled at the same time. Thus, multiple `FlexrayTpConnections` can be defined on the same ECU. Each `FlexrayTpConnection` is controlled by configuration parameters defined in `FlexrayTpConnectionControl`.

[TPS_SYST_01102] FlexrayTpConnectionControl reuse [The same `FlexrayTpConnectionControl` may be reused for an arbitrary number of `FlexrayTpConnections`.] [\(RS_SYST_00014\)](#)

A `FlexrayTpConnection` defines the way of communication between a sender and a receiver and uses a `FlexrayTpPduPool` of `NPdu`s to transmit data to the FlexRay Interface.

[TPS_SYST_01103] `FlexrayTpConnection` shall specify one `txPduPool` [Each `FlexrayTpConnection` shall specify one `txPduPool` with at least one `nPdu`.] (*RS_SYST_00014*)

In order to achieve a higher bandwidth a `txPduPool` may contain more than one transmit `NPdu`, e.g. if all referenced `NPdu`s are transmitted in different `FlexrayFrames` in the same cycle.

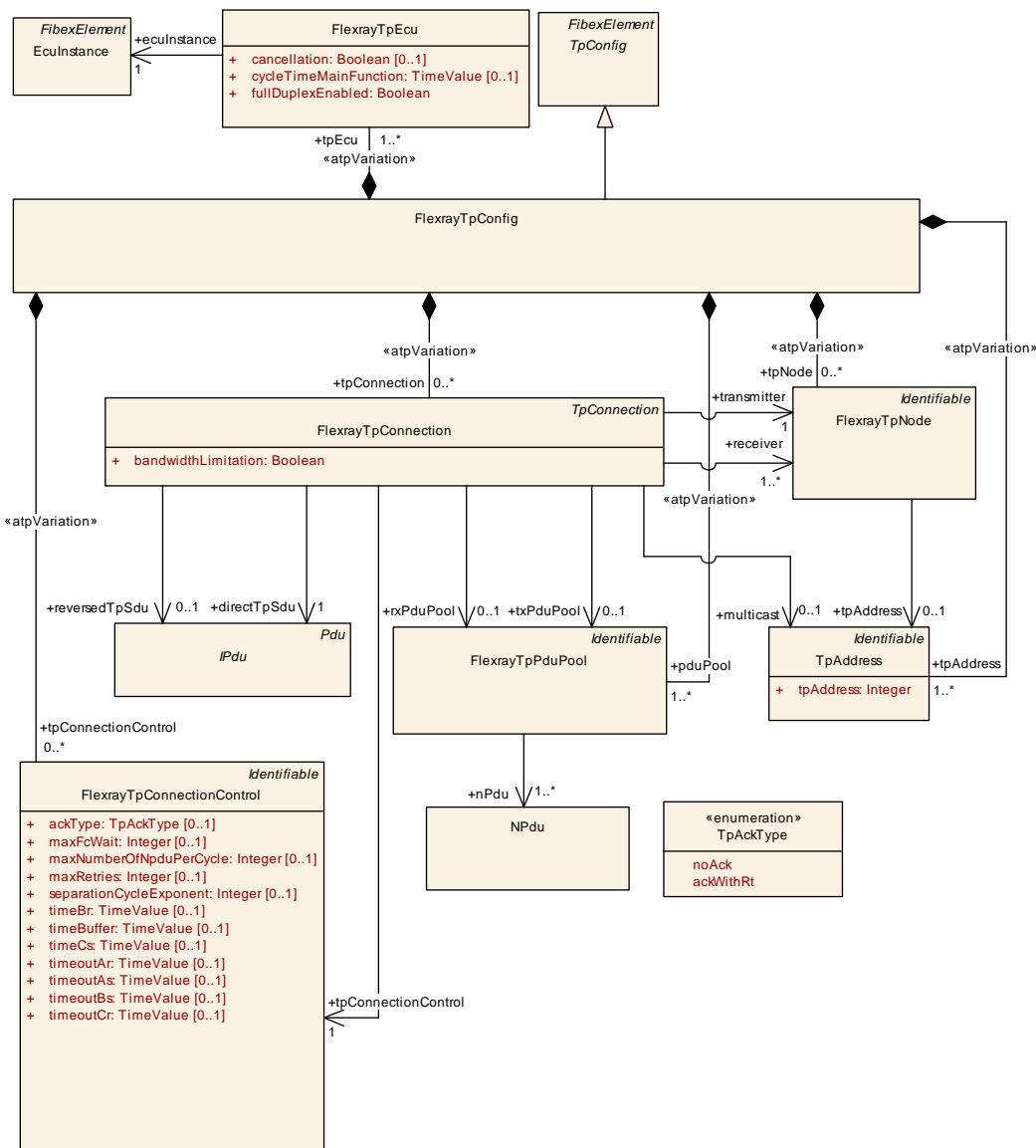


Figure 6.56: FlexRay ISO Transport Layer Configuration (TransportProtocols: FlexRay-IsoTransportProtocol)

`FlexrayTpConnections` are specifically used for communication between one source and one or several target devices. These communication partners are specified

using the [transmitter](#) and [receiver](#) associations to [FlexrayTpNodes](#), providing the diagnostic [tpAddress](#) and the connection to the topology.

[TPS_SYST_01104] FlexrayTpConnection with several receivers [In case of several receivers a multicast [tpAddress](#) shall be used.] ([RS_SYST_00014](#))

The actual payload to be transported by the [FlexrayTpConnection](#) is specified by using either one or two references to [IPdu](#)s, depending on whether the connection shall be used unidirectional (one reference) or bidirectional (two references).

Class	FlexrayTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	<p>This element defines exactly one FlexRay ISO TP Configuration.</p> <p>One FlexRayTpConfig element shall be created for each FlexRay Network in the System that uses Flex Ray Iso Tp.</p> <p>Tags:atp.recommendedPackage=TpConfigs</p>			
Base	ARObject , CollectableElement , FibexElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , TpConfig			
Attribute	Type	Mult.	Kind	Note
pduPool	FlexrayTpPduPool	1..*	aggr	<p>Configuration of FlexRay TP Pdu Pools.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpAddress	TpAddress	1..*	aggr	<p>Collection of TpAddresses.</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpConnection	FlexrayTpConnection	*	aggr	<p>Configuration of FlexRay TP Connections.</p> <p>atpVariation: Derived, because TpNode can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpConnection Control	FlexrayTpConnection Control	*	aggr	<p>Configuration of FlexRay TP Connection Controls.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpEcu	FlexrayTpEcu	1..*	aggr	<p>Collection of TP Ecus</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpNode	FlexrayTpNode	*	aggr	<p>Senders and receivers of FlexRay TP messages.</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>

Table 6.206: FlexrayTpConfig

Class	FlexrayTpConnectionControl			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Configuration parameters to control a FlexRay TP connection.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
ackType	TpAckType	0..1	attr	This parameter defines the type of acknowledgement which is used for the specific channel.
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of Flow Control N-PDUs with FlowState "WAIT".
maxNumberOfNpduPerCycle	Integer	0..1	attr	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.
maxRetries	Integer	0..1	attr	This parameter defines the maximum number of retries (if retry is configured for the particular channel).
separationCycleExponent	Integer	0..1	attr	Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.
timeBr	TimeValue	0..1	attr	Time (in seconds) until transmission of the next Flow Control N-PDU.
timeBuffer	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to get a Tx or Rx buffer. This parameter is equivalent to the temporal distance between two FC.WT N-Pdus in case the buffer request returns busy. Specified in seconds.
timeCs	TimeValue	0..1	attr	Time (in seconds) until transmission of the next ConsecutiveFrame NPdu / LastFrame NPdu.
timeoutAr	TimeValue	0..1	attr	This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the Flex Ray Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the Flex Ray Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.

Table 6.207: FlexrayTpConnectionControl

Class	FlexrayTpConnection
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols





Class		FlexrayTpConnection		
Note	<p>A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection.</p> <p>In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.</p>			
Base	<i>ARObject, TpConnection</i>			
Attribute	Type	Mult.	Kind	Note
bandwidth Limitation	Boolean	1	attr	Specifies whether the connection requires a bandwidth limitation or not.
directTpSdu	IPdu	1	ref	Reference to the IPdu that is segmented by the Transport Protocol.
multicast	TpAddress	0..1	ref	TP address for 1:n connections.
receiver	FlexrayTpNode	1..*	ref	The target of the TP connection.
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction.
rxPduPool	FlexrayTpPduPool	0..1	ref	<p>A connection has a reference to a set of NPdus (FrTpRx PduPool) which are defined for receiving data via this particular connection.</p> <p>The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the rxPduPool holds the actually received NPdus. In case this connection is applied to the receiver the rxPduPool holds the actually sent NPdus.</p>
tpConnection Control	FlexrayTpConnection Control	1	ref	Reference to the connection control.
transmitter	FlexrayTpNode	1	ref	The source of the TP connection.
txPduPool	FlexrayTpPduPool	0..1	ref	<p>A connection has a reference to a set of NPdus (FrTpTx PduPool) which are defined for sending data via this particular connection.</p> <p>The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the txPduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus.</p>

Table 6.208: FlexrayTpConnection

The [FlexrayTpConnection](#) refers to the [FlexrayTpPduPool](#) in two roles: [rxPduPool](#) and [txPduPool](#).

[TPS_SYST_01064] Transmit/Receive Semantics of Pdu Pools [The transmit/receive semantics of Pdu Pools depends on the role of the regarded ECU:

- If the ECU is the transmitter then the [txPduPool](#) holds the sent NPdus and the [rxPduPool](#) holds the received NPdus.
- If the ECU is the receiver then the the [txPduPool](#) holds the received NPdus and the [rxPduPool](#) holds the sent NPdus.

]()

The following example shows how this differentiation may be used:

System Description:

SENDER = A
RECEIVER = B
TxPool = PDU_1
RxPool = PDU_2

ECU Extract of A:

SENDER = A
TxPool = PDU_1 -> sent Pdus
RxPool = PDU_2 -> received Pdus

Since on receiver side the PDU_1 is received and PDU_2 is sent (from a local point of view) the export shall look like this:

ECU Extract of B:

RECEIVER = B
TxPool = PDU_1 -> received Pdus
RxPool = PDU_2 -> sent Pdus

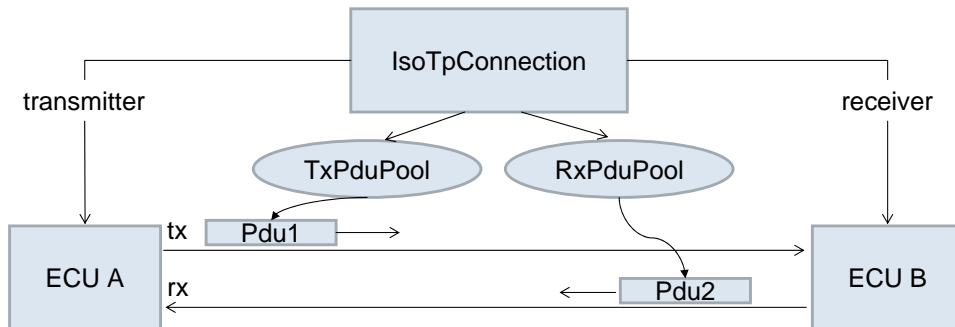


Figure 6.57: IsoTp Example

Class	FlexrayTpPduPool			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.			
Base	ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
nPdu	NPdu	1..*	ref	Reference to NPdus that are part of the PduPool.

Table 6.209: FlexrayTpPduPool

Class	FlexrayTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
connector	Communication Connector	*	ref	Association to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.210: FlexrayTpNode

Class	FlexrayTpEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
cycleTimeMain Function	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstance	EcuInstance	1	ref	Connection to the ECUInstance in the Topology
fullDuplex Enabled	Boolean	1	attr	The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.

Table 6.211: FlexrayTpEcu

6.8.3 FlexRay AUTOSAR Transport Layer

This section describes a Non-ISO FlexRay TP protocol that is supported by AUTOSAR in addition to the FlexRay ISO 10681-2 TP (see section 6.8.2). The Non-ISO FlexRay Transport Layer supports multiple sessions, i.e. multiple segmented transfers can be handled at the same time.

A `FlexrayArTpChannel` provides a Tx and an Rx pool of `NPdus` which are used by the associated `FlexrayArTpConnections`.

`FlexrayArTpConnections` are used for communication between one `source` and one or more `target` device(s). These communication partners are specified by the `source` and `target` associations to `FlexrayArTpNodes`, providing the diagnostic `TpAddresses` and the connection to the topology description. The actual payload to be transported by the `FlexrayArTpConnection` is identified by the references `directTpSdu` and `reversedTpSdu` to `IPdus`. When one of the two SDUs is omitted, the connection shall be used unidirectional.

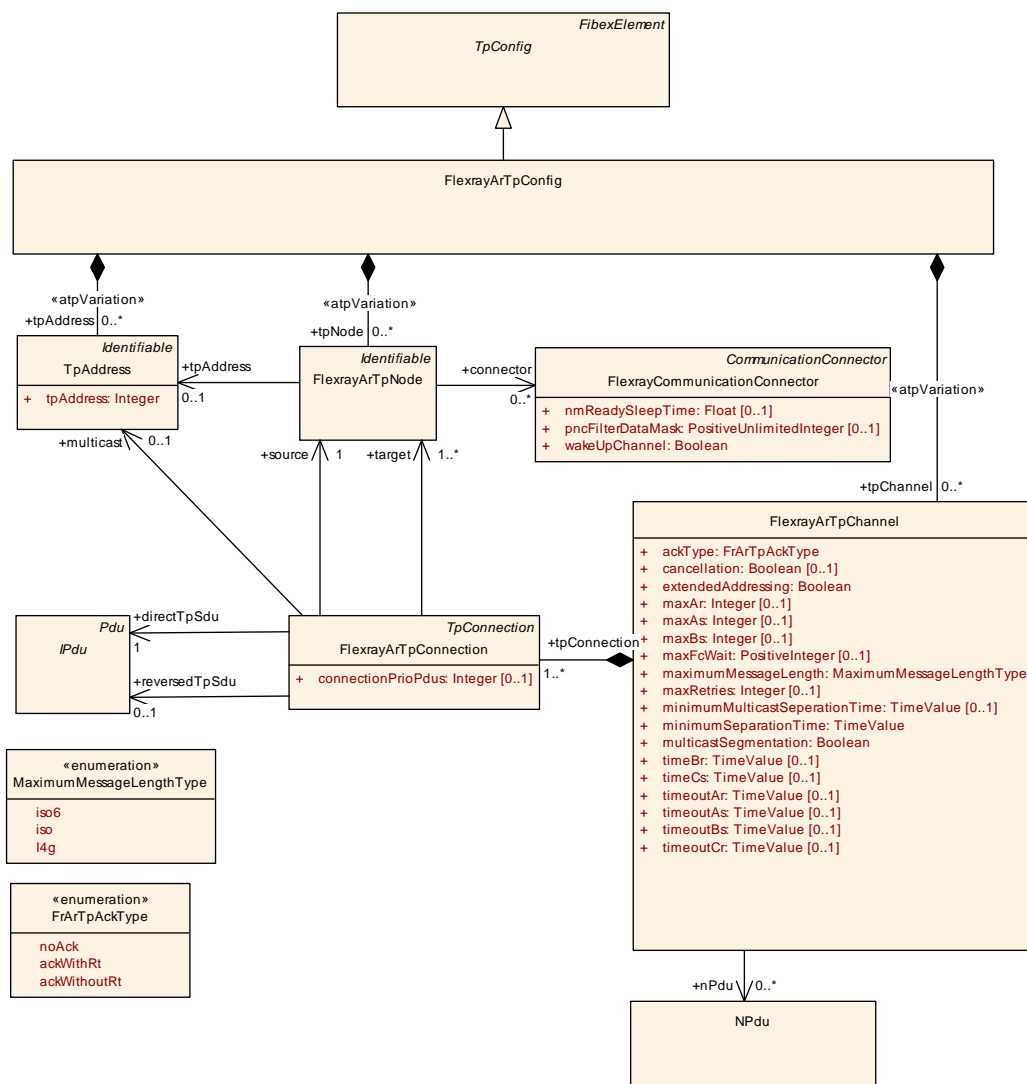


Figure 6.58: FlexRay Autosar Transport Layer Configuration (TransportProtocols: FlexRayAutosarTransportProtocol)

Class	FlexrayArTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	<p>This element defines exactly one FlexRay Autosar TP Configuration.</p> <p>One FlexrayArTpConfig element shall be created for each FlexRay Network in the System that uses Flex Ray Autosar TP.</p> <p>Tags:atp.recommendedPackage=TpConfigs</p>			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable, TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
tpAddress	TpAddress	*	aggr	<p>Collection of TpAddresses.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpChannel	FlexrayArTpChannel	*	aggr	<p>Configuration of FlexRay Autosar Transport Protocol channels.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
tpNode	FlexrayArTpNode	*	aggr	<p>Senders and receivers of TP messages.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>

Table 6.212: FlexrayArTpConfig

Class	FlexrayArTpChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	<p>A channel is a group of connections sharing several properties.</p> <p>The FlexRay AutosarTransport Layer supports several channels. These channels can work concurrently, thus each of them requires its own state machine and management data structures and its own PDU-IDs.</p>			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
ackType	FrArTpAckType	1	attr	Type of Acknowledgement.
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
extended Addressing	Boolean	1	attr	Adressing Type of this connection: true: Two Bytes false: One Byte
maxAr	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
maxAs	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured).
maxBs	Integer	0..1	attr	This attribute defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.
maxFcWait	PositiveInteger	0..1	attr	This attribute defines the maximal number of wait frames to be sent for a pending connection. Range is 0..255.
maximum MessageLength	MaximumMessageLengthType	1	attr	This specifies the maximum message length for the particular channel.





Class	FlexrayArTpChannel			
maxRetries	Integer	0..1	attr	This attribute defines the maximum number of retries (if retry is configured for the particular channel).
minimum Multicast SeparationTime	TimeValue	0..1	attr	<p>This attribute defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>minimumMulticastSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $\text{minimumMulticastSeparationTime} = n * \text{cycle} * m$, where n is an integer ≥ 0, cycle is Flexray Cluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrTp, minimumMulticastSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p> <p>Range: 0 .. 0.127</p>
minimum SeparationTime	TimeValue	1	attr	<p>This attribute defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>The minimumSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $\text{minimumSeparationTime} = n * \text{cycle} * m$, where n is an integer ≥ 0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.</p> <p>Please note: Due to the scheduling strategies of FrTp, minimumSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p> <p>Range: 0 .. 0.127</p>
multicast Segmentation	Boolean	1	attr	This attribute defines whether segmentation within a 1:n connection is allowed or not.
nPdu	NPdu	*	ref	A FlexRayTpChannel references a set of NPdus. These NPdus are logically assembled into a pool of Rx NPdus and another pool of Tx NPdus. It shall be ensured that a second channel either references all NPdus of such a pool, or none.
timeBr	TimeValue	0..1	attr	This attribute defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.
timeCs	TimeValue	0..1	attr	This attribute defines the time in seconds between the sending of two consecutive frames or between a consecutive frame and a flow control (for Transmit Cancellation) or between reception of a flow control or Acknowledgement Frame and sending of the next consecutive frame or a flow control (for Transmit Cancellation).





Class	FlexrayArTpChannel			
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request of the Transport Layer to the Flex Ray Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).
timeoutBs	TimeValue	0..1	attr	This attribute defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This attribute defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.
tpConnection	FlexrayArTpConnection	1..*	aggr	Group of connections that can be used in this channel.

Table 6.213: FlexrayArTpChannel

Class	FlexrayArTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
connector	FlexrayCommunicationConnector	*	ref	Association to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.214: FlexrayArTpNode

Class	FlexrayArTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection within a channel identifies the sender and the receiver of this particular communication. The FlexRay Autosar Tp module routes a Pdu through this connection.			
Base	<i>ARObject</i> , <i>TpConnection</i>			
Attribute	Type	Mult.	Kind	Note
connectionPrioPdus	Integer	0..1	attr	This parameter defines the number of PDUs that shall be reserved for this connection when it is active. The range is 1-255.





Class	FlexrayArTpConnection			
directTpSdu	IPdu	1	ref	Reference to the IPdu that is segmented by the Transport Protocol. The source address of the transmitted NPdu is determined by the configured source Communication Connector. The target address of the transmitted NPdu is determined by the configured target Communication Connector.
multicast	TpAddress	0..1	ref	TP address for 1:n connections.
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction. The source address of the transmitted NPdu is determined by the configured target Communication Connector. The target address of the transmitted NPdu is determined by the configured source Communication Connector.
source	FlexrayArTpNode	1	ref	The source of the TP connection.
target	FlexrayArTpNode	1..*	ref	The target of the TP connection.

Table 6.215: FlexrayArTpConnection

Enumeration	FrArTpAckType
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Type of Acknowledgement.
Literal	Description
ackWithoutRt	Acknowledgement without retry. Tags: atp.EnumerationLiteralIndex=0
ackWithRt	Acknowledgement with retry. Tags: atp.EnumerationLiteralIndex=1
noAck	No acknowledgement. Tags: atp.EnumerationLiteralIndex=2

Table 6.216: FrArTpAckType

Enumeration	MaximumMessageLengthType
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Type of Acknowledgement.
Literal	Description
l4g	SF-E allowed (SF of arbitrary length depending on FrTpPduLength), up to (2**32)-1 byte message length (all FF-x allowed). Tags: atp.EnumerationLiteralIndex=0
iso	Up to (2**12)-1 Byte message length (No FF-Ex or SF-E or AF shall be used and recognized). Tags: atp.EnumerationLiteralIndex=1





<i>Enumeration</i>	MaximumMessageLengthType
iso6	As ISO, but the maximum payload length is limited to 6 byte (SF-I, FF-I, CF). This is necessary to route TP on CAN when using Extended Addressing or Mixed Addressing on CAN. Tags: atp.EnumerationLiteralIndex=2

Table 6.217: MaximumMessageLengthType

6.8.4 CAN Transport Layer

The CAN Transport Layer supports multiple sessions by means of `CanTpChannels`: Each `CanTpChannel` uses its own resources, such as internal buffer, timer, state machine and thus can operate independently and simultaneously to other `CanTpChannels`. The same session can be reused for an arbitrary number of `CanTpConnections`.

Each `CanTpConnection` uses its own pair of `NPdu`: One `NPdu`, the `dataPdu` is mandatory for each `CanTpConnection`, the `flowControlPdu` is optional depending whether only Single Frames are transferred over the connection.

A `CanTpConnection` is specifically used for communication between source and target devices. These communication partners are specified using the `transmitter` and `receiver` associations to `CanTpNode`, providing the diagnostic `tpAddress` and the connection to the topology.

[TPS_SYST_01146] Generic `CanTpConnections` [If the `transmitter` or the `receiver` of a `CanTpConnection` is not specified then the `CanTpConnection` is a generic one (address information is not determined).]()

[TPS_SYST_01105] `CanTpConnection` with several receivers [In case of several receivers a multicast `tpAddress` shall be used.] (*RS_SYST_00014*)

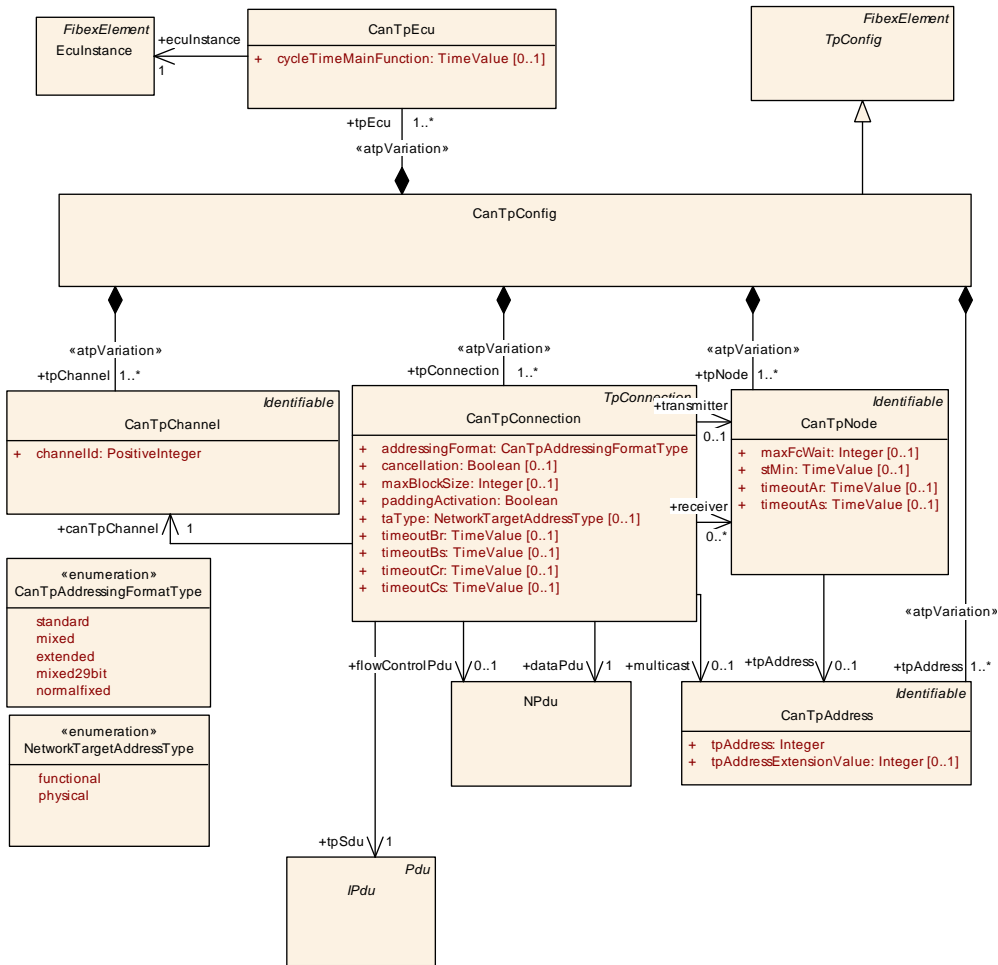


Figure 6.59: CAN Transport Layer Configuration (TransportProtocols: CanTransportProtocol)

The actual payload to be transported by the `CanTpConnection` is specified by the reference `tpSdu` to `IPdu`.

The `N_TAtype` communication models as defined in ISO 15765-2 [26] can be expressed using a combination of the attributes `addressingFormat` (`CanTpAddressingFormatType`) and `taType` (`NetworkTargetAddressType`).

Class	CanTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	<p>This element defines exactly one CAN TP Configuration.</p> <p>One CanTpConfig element shall be created for each CAN Network in the System.</p> <p>Tags:atp.recommendedPackage=TpConfigs</p>			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
tpAddress	CanTpAddress	1..*	aggr	<p>Collection of TP Adresses.</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=postBuild</p>
tpChannel	CanTpChannel	1..*	aggr	<p>Configuration of CAN TP channels.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=postBuild</p>
tpConnection	CanTpConnection	1..*	aggr	<p>Senders and receivers of CAN TP messages.</p> <p>atpVariation: Derived, because TpNode can vary.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=postBuild</p>
tpEcu	CanTpEcu	1..*	aggr	<p>Collection of TP Ecus</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=postBuild</p>
tpNode	CanTpNode	1..*	aggr	<p>Senders and receivers of Can TP messages.</p> <p>atpVariation: Derived, because EcuInstance can vary.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=postBuild</p>

Table 6.218: CanTpConfig

Class	CanTpChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Configuration parameters of the CanTp channel.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
channelId	PositiveInteger	1	attr	The id of the channel. The value shall be unique for each channel.

Table 6.219: CanTpChannel

Class	CanTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.			
Base	ARObject, TpConnection			
Attribute	Type	Mult.	Kind	Note
addressingFormat	CanTpAddressingFormatType	1	attr	Declares which communication addressing mode is supported.
cancellation	Boolean	0..1	attr	With this switch Tx Cancellation can be turned on or off. Please note that the Rx Cancellation is always enabled.
canTpChannel	CanTpChannel	1	ref	Reference to the CanTpChannel on which this CanTp Connection is realized.
dataPdu	NPdu	1	ref	Reference to an Data NPdu.
flowControlPdu	NPdu	0..1	ref	Reference to the Flow Control NPdu.
maxBlockSize	Integer	0..1	attr	The maximum number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification. Note: For reasons of buffer length, the CAN Transport Layer can adapt the BS value within the limit of this maximum BS
multicast	CanTpAddress	0..1	ref	TP address for 1:n connections.
paddingActivation	Boolean	1	attr	This specifies wheter or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload. true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes) false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)
receiver	CanTpNode	*	ref	The target of the TP connection.
taType	NetworkTargetAddressType	0..1	attr	Network Target Address type.
timeoutBr	TimeValue	0..1	attr	Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.





<i>Class</i>	CanTpConnection			
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
timeoutCs	TimeValue	0..1	attr	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
tpSdu	IPdu	1	ref	Reference to an IPdu that is segmented by the Transport Protocol.
transmitter	CanTpNode	0..1	ref	The source of the TP connection.

Table 6.220: CanTpConnection

Enumeration	CanTpAddressingFormatType
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Declares which communication addressing mode is supported.
Literal	Description
extended	To use extended addressing format. Tags: atp.EnumerationLiteralIndex=0
mixed	To use mixed 11bit addressing format. Tags: atp.EnumerationLiteralIndex=1
mixed29bit	To use mixed 29bit addressing format Tags: atp.EnumerationLiteralIndex=2
normalfixed	To use normal fixed addressing format Tags: atp.EnumerationLiteralIndex=3
standard	To use normal addressing format. Tags: atp.EnumerationLiteralIndex=4

Table 6.221: CanTpAddressingFormatType

Class	CanTpAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	<i>ARObject</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Attribute	Type	Mult.	Kind	Note
tpAddress	Integer	1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.
tpAddress ExtensionValue	Integer	0..1	attr	If the mixed addressing format is used, this parameter contains the transport protocol address extension value.

Table 6.222: CanTpAddress

Class	CanTpEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
cycleTimeMain Function	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstance	EcuInstance	1	ref	Connection to the ECUInstance in the Topology

Table 6.223: CanTpEcu

Class	CanTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
connector	CommunicationConnector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.
stMin	TimeValue	0..1	attr	Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
tpAddress	CanTpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.224: CanTpNode

Enumeration	NetworkTargetAddressType
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Network Target Address type (see ISO 15765-2).
Literal	Description
functional	Functional request type Tags: atp.EnumerationLiteralIndex=0
physical	Physical request type Tags: atp.EnumerationLiteralIndex=2

Table 6.225: NetworkTargetAddressType

6.8.5 LIN Transport Layer

`LinTpConnection` is used for modeling communication resources required for using the LIN Transport Layer. Contrary to the FlexRay and CAN Transport Layers, LIN TP only supports one session per `PhysicalChannel`.

An arbitrary number of `LinTpConnections` per `LinTpConfig` can be defined since the transmission of data from master to slave, using the `MasterRequest` frame, and the transmission of data from slave to master, using the `SlaveResponse` frame, needs to be described per NAD the `LinMaster` uses to address one or more of its `LinSlaves`.

`LinTpConnection` uses the `dataPdu` reference for specifying exactly one `NPdu` which is to be used for transmitting the data, and it optionally references a `flow-Control NPdu` in order to handle Flow Control Frames if required.

One `LinTpConnection` is specifically used for communication between one source and one or several target devices. These communication partners are specified using the `transmitter` and `receiver` associations to `LinTpNode`, providing the diagnostic `tpAddress` and the connection to the topology. In case of several receivers a `multicast tpAddress` shall be used.

The actual payload to be transported by the `LinTpConnection` is specified by the reference `linTpNSdu` to `IPdu`.

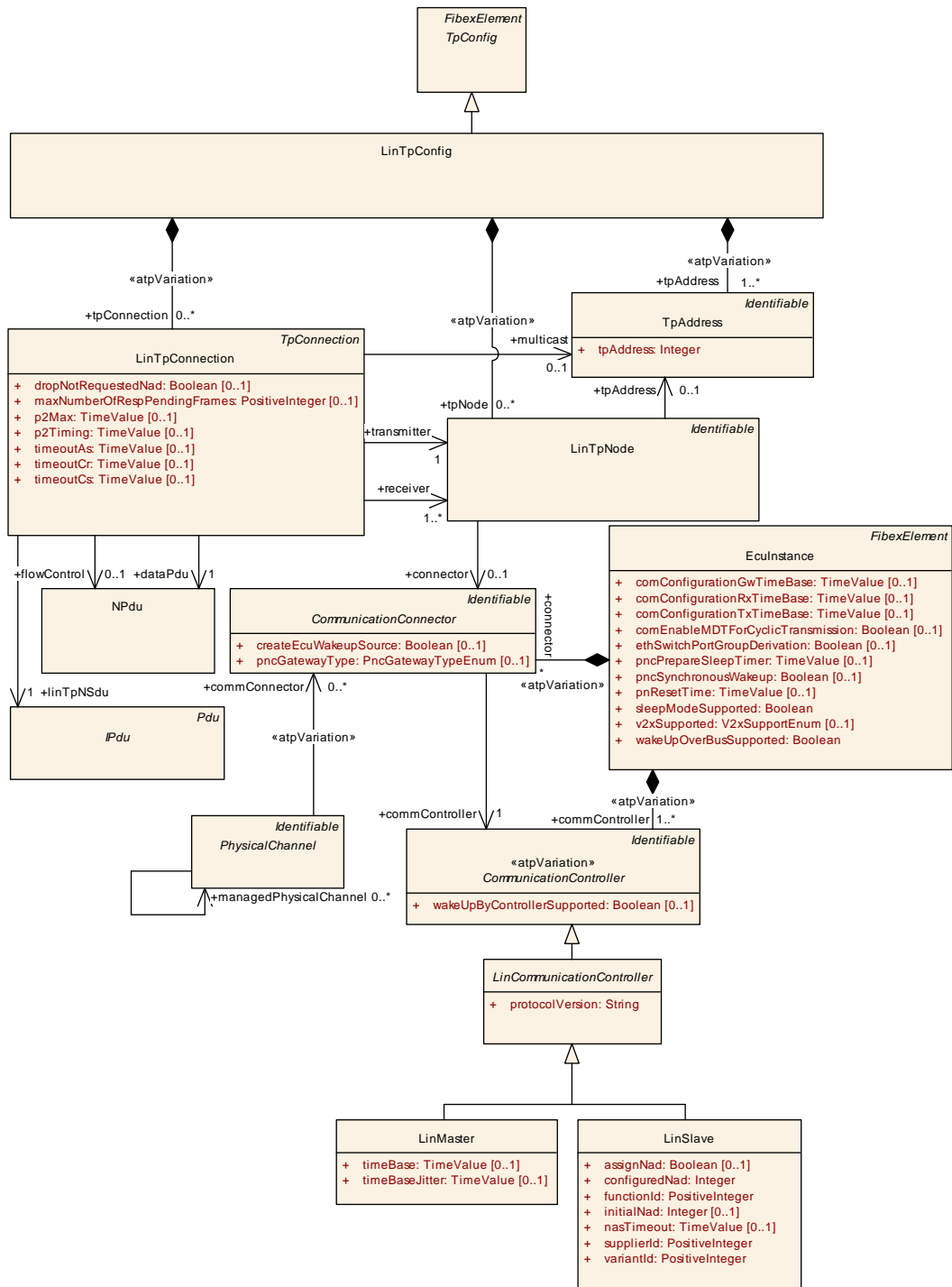


Figure 6.60: LIN Transport Layer Configuration

Class	LinTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one Lin TP Configuration. One LinTpConfig element shall be created for each Lin Network in the System. Tags: atp.recommendedPackage=TpConfigs			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
tpAddress	TpAddress	1..*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
tpConnection	LinTpConnection	*	aggr	Configuration of LIN TP channels. atpVariation: Derived, because TpNode can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
tpNode	LinTpNode	*	aggr	Senders and receivers of LIN TP messages. atpVariation: Derived, because EcuInstance can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.226: LinTpConfig

Class	LinTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
connector	CommunicationConnector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.227: LinTpNode

Class	LinTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			





Class	LinTpConnection			
Note	<p>A LinTP channel represents an internal path for the transmission or reception of a Pdu via LinTp and describes the sender and the receiver of this particular communication.</p> <p>LinTp supports (per Lin Cluster) the configuration of one Rx Tp-SDU and one Tx Tp-SDU per NAD the LinMaster uses to address one or more of its Lin Slaves. To support this an arbitrary number of LinTp Connections shall be described.</p>			
Base	<i>ARObject, TpConnection</i>			
Attribute	Type	Mult.	Kind	Note
dataPdu	NPdu	1	ref	<p>Reference to an NPdu (Single Frame, First Frame or Consecutive Frame).</p> <p>The Single Frame network protocol data unit (SF N_PDU) shall be sent out by the sending network entity and can be received by one or multiple receiving network entities. The Single Frame (SF N_PDU) shall be sent out to transfer a service data unit that can be transferred via a single service request to the data link layer. This network protocol data unit shall be sent to transfer unsegmented messages.</p> <p>The First Frame network protocol data unit (FF N_PDU) identifies the first network protocol data unit (N_PDU) of a segmented message transmitted by a network sending entity and received by a receiving network entity.</p> <p>The Consecutive Frame network protocol data unit (CF N_PDU) transfers segments (N_Data) of the service data unit message data (<MessageData>). All network protocol data units (N_PDU) transmitted by the sending entity after the First Frame network protocol data unit (FF N_PDU) shall be encoded as Consecutive Frames network protocol data units (CF N_PDU).</p>
dropNotRequestedNad	Boolean	0..1	attr	Configures if TP Frames of not requested LIN-Slaves are dropped or not.
flowControl	NPdu	0..1	ref	<p>Reference to the Flow Control NPdu.</p> <p>The Flow Control network protocol data unit (FC N_PDU) is identified by the Flow Control protocol control information (FC N_PCI). The Flow Control network protocol data unit (FC N_PDU) instructs a sending network entity to start, stop or resume transmission of CF N_PDU. The Flow Control network protocol data unit shall be sent by the receiving network layer entity to the sending network layer entity, when ready to receive more data, after correct reception of:</p> <p>a) First Frame network protocol data unit (FF N_PDU)</p> <p>b) the last Consecutive Frame network protocol data unit (CF N_PDU) of a block of Consecutive Frames (CF N_PDU) if further Consecutive Frame network protocol data unit (CF N_PDU) need(s) to be sent.</p>
linTpNSdu	IPdu	1	ref	Reference to the IPdu that is segmented by the Transport Protocol.
maxNumberOfRespPendingFrames	PositiveInteger	0..1	attr	Configures the maximum number of allowed response pending frames.
multicast	TpAddress	0..1	ref	TP address for 1:n connections.
p2Max	TimeValue	0..1	attr	After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.





Class	LinTpConnection			
p2Timing	TimeValue	0..1	attr	P2 timeout observation parameter.
receiver	LinTpNode	1..*	ref	The target of the TP connection.
timeoutAs	TimeValue	0..1	attr	Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This attribute defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
timeoutCs	TimeValue	0..1	attr	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
transmitter	LinTpNode	1	ref	The source of the TP connection.

Table 6.228: LinTpConnection

6.8.6 Ethernet Transport Layer

The Transport Layer in the AUTOSAR Ethernet protocol stack is defined by the Tcplp module. For the transmission of an upper layer module Pdu via an UDP or TCP socket, the AUTOSAR Socket Adaptor specifies a Pdu route which is linked to a socket connection. The upper layer module of the SoAd may use the Interface (IF) API or the Transport Protocol (TP) API for the transmit request and data provision respectively.

With the `EthTpConnection` it is possible to describe in a System Description that the TP API shall be used for a specific Pdu route. If a `PduTriggering` is not referenced by a `EthTpConnection` the IF API will be used.

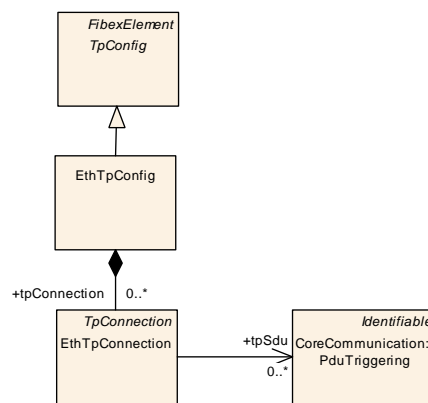


Figure 6.61: Modeling of EthTpConnection

Class	EthTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines which PduTriggerings shall be handled using "TP" semantics. Tags: atp.recommendedPackage=TpConfigs			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, TpConfig			
Attribute	Type	Mult.	Kind	Note
tpConnection	EthTpConnection	*	aggr	Senders and receivers of SOME/IP TP messages.

Table 6.229: EthTpConfig

Class	EthTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection identifies which PduTriggerings shall be handled using the "TP" semantics.			
Base	ARObject, TpConnection			
Attribute	Type	Mult.	Kind	Note
tpSdu	PduTriggering	*	ref	Reference to a PduTriggering that shall be transported using the "TP" semantics.

Table 6.230: EthTpConnection

6.8.7 SOME/IP segmenter

On the transmission side SOME/IP TP segments an incoming SOME/IP IPdu that does not fit into a single UDP Package into smaller *GeneralPurposeIPdus* with category *SOMEIP_SEGMENTED_IPDU* and allows to transport SOME/IP messages over UDP that are greater than 128KB. On the reception side the large IPdu is reassembled again. The Message Type field of the SOME/IP header contains a bit, which marks the SOME/IP message as a segment of an original SOME/IP message. Every segmented SOME/IP message adds SOME/IP TP specific fields to the SOME/IP header. These fields contain control information for the segmentation and the reassembly of original, large SOME/IP messages.

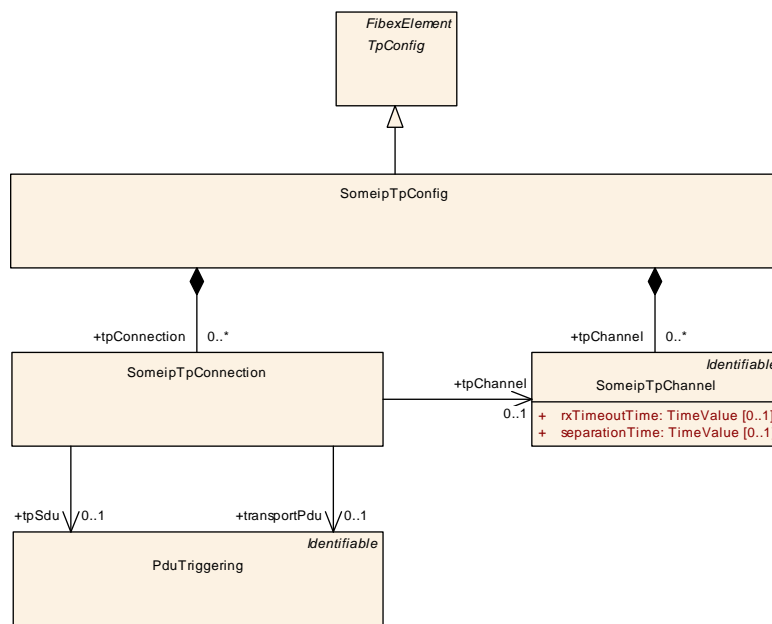


Figure 6.62: SOME/IP Segmenter

Class	SomeipTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one SOME/IP TP Configuration. Tags: atp.recommendedPackage=TpConfigs			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> , <i>TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
tpChannel	SomeipTpChannel	*	aggr	Definition of SomeipTpChannels that are collecting configuration properties that are valid for a collection of SomeipTpConnections.
tpConnection	SomeipTpConnection	*	aggr	Senders and receivers of SOME/IP TP messages.

Table 6.231: SomeipTpConfig

Class	SomeipTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection identifies the sender and the receiver of this particular communication. The SOME/IP TP module routes a Pdu through this connection.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
tpChannel	SomeipTpChannel	0..1	ref	Assignment of configuration properties valid for this SomeipTpConnection.
tpSdu	PduTriggering	0..1	ref	Reference to an IPdu that is segmented by the Transport Protocol.
transportPdu	PduTriggering	0..1	ref	Reference to the segmented IPdu.

Table 6.232: SomeipTpConnection

Class	SomeipTpChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element is used to assign properties to SomeipTpConnections that are referencing this SomeipTp Channel.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
rxTimeoutTime	TimeValue	0..1	attr	Timer to monitor the successful reception. It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received.
separationTime	TimeValue	0..1	attr	Sets the duration of the minimum time in seconds the SOME/IP TP module shall wait between the transmissions of NPdus.

Table 6.233: SomeipTpChannel

[constr_3328] [SomeipTpConnection.transportPdu](#) reference restriction [A [PduTriggering](#) that is referenced by a [SomeipTpConnection](#) in the role `transportPdu` shall reference a [GeneralPurposeIPdu](#) with category `SOMEIP_SEGMENTED_IPDU` in the role `iPdu`.]()

[constr_3329] [SomeipTpConnection.tpSdu](#) reference restriction [A [PduTriggering](#) that is referenced by a [SomeipTpConnection](#) in the role `tpSdu` shall reference an [IPdu](#) in the role `iPdu`.]()

[TPS_SYST_02156] Length of [GeneralPurposeIPdu](#) with category `SOMEIP_SEGMENTED_IPDU` [The `length` of [GeneralPurposeIPdu](#) with category `SOMEIP_SEGMENTED_IPDU` that is referenced by a [PduTriggering](#) in the role `iPdu` that in turn is referenced by a [SomeipTpConnection](#) in the role `transportPdu` defines the maximum size in bytes of a segment.]([RS_SYST_00050](#), [RS_SYST_00039](#), [RS_SYST_00014](#))

Please note that the `length` of a [GeneralPurposeIPdu](#) with category `SOMEIP_SEGMENTED_IPDU` covers 8 bytes of the SOME/IP header, 4 bytes of the TP header, and the segment itself.

[constr_3330] Same `transportPdu` shall not be used in different [SomeipTpConnections](#) [A [PduTriggering](#) that is referencing a [GeneralPurposeIPdu](#) with

category SOMEIP_SEGMENTED_IPDU in the role `iPdu` shall be referenced at most once by a `SomeipTpConnection` in the role `transportPdu`.|()

6.8.8 SAE J1939 Transport Layer

There are two transport protocol variants defined by J1939: BAM (Broadcast Announce Message), which is a broadcast protocol that does not use any flow control, and CMDT (Connection Mode Data Transfer), which is a point-to-point protocol with flow control and acknowledgment.

BAM uses two NPdus for transport, TP.CM (Transport Protocol Command, `flowControlPdu`) and TP.DT (Transport Protocol Data, `dataPdu`). CMDT uses three NPdus, because an additional TP.CM (`flowControlPdu`) in reverse direction is needed for flow control. The length of TP.CM and TP.DT NPdus is fixed to 8 bytes.

[TPS_SYST_01106] Usage of additional `directPdu` in case of variable length `sdu`

[In case of variable length `sdu` (with system signals of variable length) an additional `directPdu` is required:

- it is used if the current length of this `sdu` is up to 8 bytes.
- if the current length of this `sdu` is higher than 8 bytes the `sdu` will be transported via the `dataPdu`.

](*RS_SYST_00014, RS_SYST_00038*)

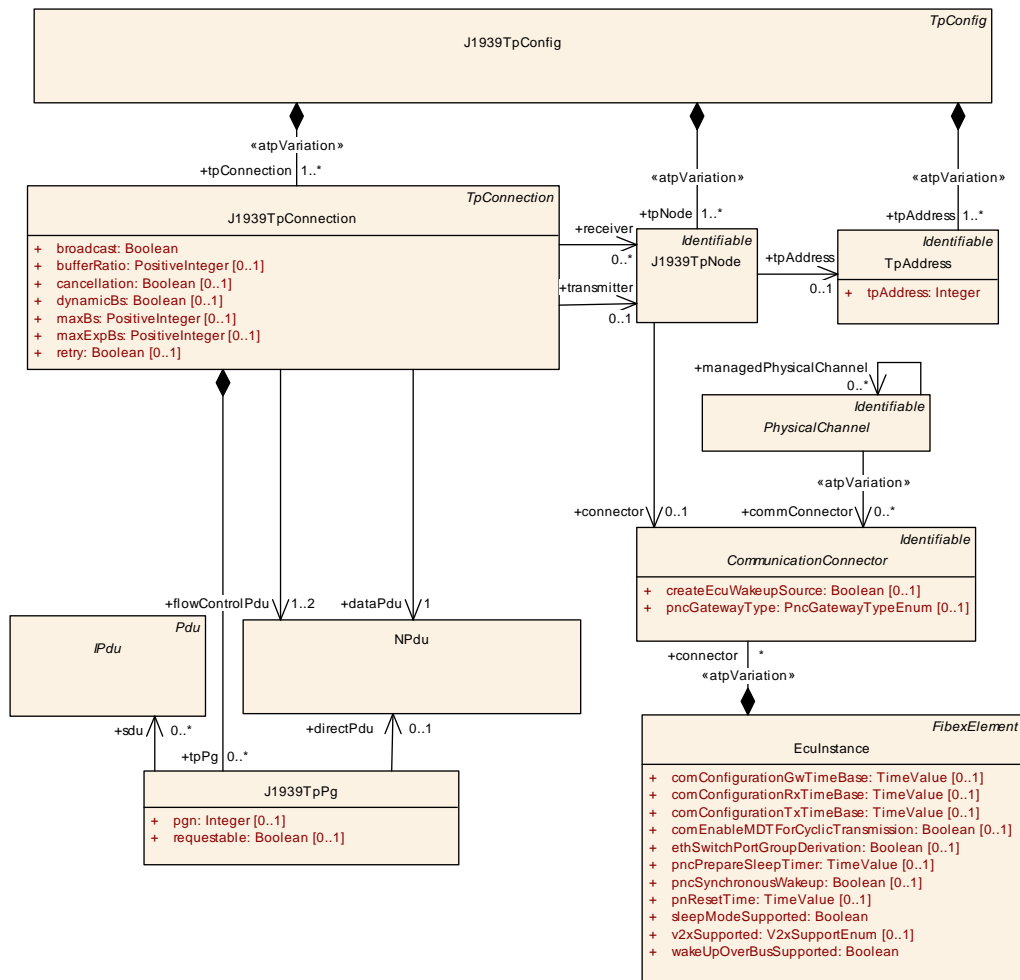


Figure 6.63: J1939 Transport Layer Configuration

A `J1939TpConnection` is specifically used for communication between source and target devices. These communication partners are specified using the `transmitter` and `receiver` associations to `J1939TpNode`, providing the diagnostic `tpAddress` and the connection to the topology.

[TPS_SYST_02190] `J1939TpConnection.transmitter` reference in case of broadcast connection [In case of a broadcast connection the `J1939TpConnection` shall only reference the `J1939TpNode` in the role `transmitter`. The reason is that BAM (Broadcast Announce Message) is always directed at the target address 0xff and therefore no `receiver` reference is necessary.] ([RS_SYST_00014](#), [RS_SYST_00038](#))

[TPS_SYST_02191] `J1939TpConnection.transmitter` reference in case that the source is an unknown node [In case that the source is an unknown node, e.g. an arbitrary tester, the `J1939TpConnection` is allowed to omit the `transmitter` reference to `J1939TpNode`.] ([RS_SYST_00014](#), [RS_SYST_00038](#))

[TPS_SYST_02192] `J1939TpConnection.receiver` reference in case that the destination is an unknown node [In case that the destination is an unknown node,

e.g. an arbitrary tester, the `J1939TpConnection` is allowed to omit the `receiver` reference to `J1939TpNode`.] (*RS_SYST_00014*, *RS_SYST_00038*)

[TPS_SYST_02193] `J1939TpConnection.receiver` reference in case that the destination is connected to a configured `J1939NmNode` [In case that the destination is connected to a configured `J1939NmNode`, the `J1939TpConnection` shall reference the `J1939TpNode` in the role `receiver`. It means that the receiving `J1939TpNode` is associated with an `EcuInstance` via the `CommunicationConnector` and this `EcuInstance` is associated with a `J1939NmNode` via the `NmEcu`. In this case the `nmNodeId` of the `J1939NmNode` corresponds to the `TpAddress` defined by `J1939TpNode`.] (*RS_SYST_00014*, *RS_SYST_00038*)

The Parameter Group (PG) to be transported by the `J1939TpConnection` is specified by the `tpPg` aggregation.

[TPS_SYST_01147] Generic `J1939TpConnections` [If the `transmitter` or the `receiver` of a `J1939TpConnection` is not specified then the `J1939TpConnection` is a generic one (address information is not determined).] ()

Class	J1939TpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one J1939 TP Configuration. One J1939TpConfig element shall be created for each J1939 Network in the System. Tags: atp.recommendedPackage=TpConfigs			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> , <i>TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
tpAddress	TpAddress	1..*	aggr	Collection of TP Adresses. atpVariation: Derived, because EcuInstance can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
tpConnection	J1939TpConnection	1..*	aggr	Configuration of J1939 TP connections. atpVariation: Derived, because TpNode can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
tpNode	J1939TpNode	1..*	aggr	Senders and receivers of J1939 TP messages. atpVariation: Derived, because EcuInstance can vary. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild

Table 6.234: J1939TpConfig

Class	J1939TpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.			
Base	<i>ARObject</i> , <i>TpConnection</i>			
Attribute	Type	Mult.	Kind	Note
broadcast	Boolean	1	attr	BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.
bufferRatio	PositiveInteger	0..1	attr	Defines usage of available data for dynamic block size calculation when protocol retry is enabled. This attribute describes in percent of available buffer that shall be used for retry.
cancellation	Boolean	0..1	attr	Enable support for Tx/Rx cancellation.
dataPdu	NPdu	1	ref	Data Message (TP.DT) used by CMDT and BAM. The DataNPdu has a fixed length of 8 bytes.
dynamicBs	Boolean	0..1	attr	Enable support for dynamic block size calculation.
flowControlPdu	NPdu	1..2	ref	Reference to the Command NPdus (TP.CM) that are used in the CMDT (Connection Mode Data Transfer) in both directions. BAM uses one TP.CM (Transport Protocol Command). The flowControlNPdu has a fixed length of 8 bytes. Please note that the role name "flowControlIPdu" is misleading and is kept for backward compatibility reasons.
maxBs	PositiveInteger	0..1	attr	Set maximum block size (number of packets in TP.CM_CTS).
maxExpBs	PositiveInteger	0..1	attr	Set maximum for expected block size (maximum number of packets in TP.CM_RTS).
receiver	J1939TpNode	*	ref	The target of the TP connection.
retry	Boolean	0..1	attr	Enable support for protocol retry.
tpPg	J1939TpPg	*	aggr	J1939 messages (parameter groups, PGs) that can be transferred via this connection.
transmitter	J1939TpNode	0..1	ref	The source of the TP connection.

Table 6.235: J1939TpConnection

Class	J1939TpPg			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
directPdu	NPdu	0..1	ref	In case of variable length IPdus (with system signals of variable length), an additional NPdu (with the PGN in the CAN ID) is used for messages with up to 8 bytes.





Class		J1939TpPg		
pgn	Integer	0..1	attr	Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a Can FrameTriggering with an identifier.
requestable	Boolean	0..1	attr	Parameter Group can be triggered by the J1939 request message.
sdu	IPdu	*	ref	Reference to IPdus that are segmented by the Transport Protocol. If more than one IPdu is referenced, the IPdus are used when the same PGN is received in parallel via different transport protocols (BAM, CDMT, direct) on the same J1939TpConnection.

Table 6.236: J1939TpPg

Class		J1939TpNode		
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	<i>ARObject</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Attribute	Type	Mult.	Kind	Note
connector	CommunicationConnector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional only when no TP is sent and only BAM is received.

Table 6.237: J1939TpNode

[constr_3210] J1939TpPgs with identical pgn value [For all [J1939TpPgs](#) where the attribute [pgn](#) has an identical value the attribute [requestable](#) shall also have an identical value.]()

6.8.9 Unicast TP Example

The example in Figure 6.64 illustrate the usage of the System Template TP model. In this System Description example the Sender ECU (Tester) communicates with the Receiver ECU (Diagnostic Server) via two Gateways (GW1 and GW2).

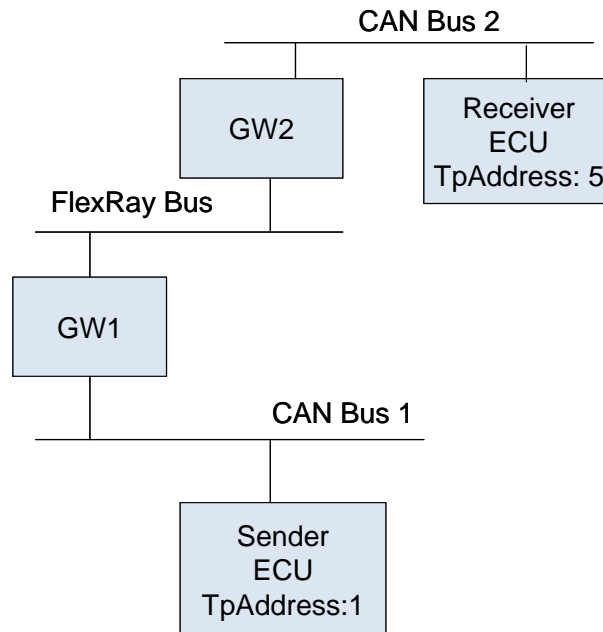


Figure 6.64: TP unicast Example

CAN Bus 1 (CanTpConfig 1):

```
CanTpConnection (CanTpConnection1):
  transmitter TpNet: Sender ECU, TpAddress: 1
  receiver TpNet: GW1, TpAddress: 5
CanTpConnection (CanTpConnection2):
  transmitter TpNet: GW1, TpAddress: 5
  receiver TpNet: Sender ECU, TpAddress: 1
```

FlexRay Bus (FlexRayTpConfig):

```
FlexRayTpConnection (FlexrayTpConnection1):
  transmitter TpNet: GW1, TpAddress: 1
  receiver TpNet: GW2, TpAddress: 5
```

CAN Bus 2 (CanTpConfig 2):

```
CanTpConnection (CanTpConnection3):
  transmitter TpNet: GW2, TpAddress: 1
  receiver TpNet: Receiver ECU, TpAddress: 5
CanTpConnection (CanTpConnection4):
  transmitter TpNet: Receiver ECU, TpAddress: 5
  receiver TpNet: GW2, TpAddress: 1
```

DiagnosticConnection:

```
physicalRequest TpConnection: CanTpConnection3
response TpConnection: CanTpConnection4
```

Please note that two different `CanTpConfig` elements are created for the two CAN networks. The `TpAddress` of the transmitter `TpNode` is always 1 and the `TpAddress` of the receiver `TpNode` is always 5, even in the `FlexrayTpConfig` where Gateway ECU1 communicates with Gateway ECU2. The original transmitter and the final receiver are addressed in each connection. Please note that for CanTp for each direction an own `CanTpConnection` is used.

The `DiagnosticConnection` is modeled only for the the last segment to which the Receiver ECU that represents the diagnostic server is connected.

6.8.10 Multicast TP Example

A second example illustrates the usage of the multicast reference.

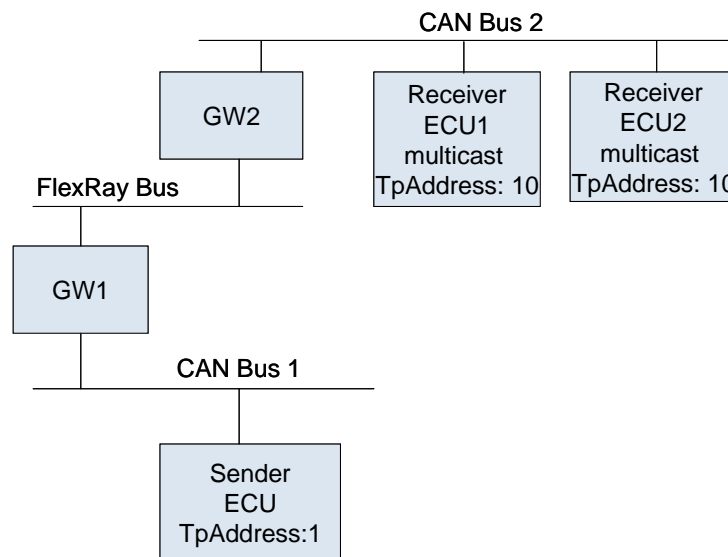


Figure 6.65: TP multicast Example

```

Can Bus 1 (CanTpConfig1):
CanTpConnection
  source TpNode: Sender ECU, TpAddress: 1
  target TpNode: GW1
  multicast TpAddress: 10
  
```

```

FlexRay Bus (FlexRayTpConfig):
FlexRayTpConnection
  source TpNode: GW1, TpAddress: 1
  target TpNode: GW2
  multicast TpAddress: 10
  
```

```

CAN Bus 2 (CanTpConfig 2):
CanTpConnectionChannel
  source TpNode: GW2, TpAddress: 1
  target TpNode: Receiver ECU1
  target TpNode: Receiver ECU2
  
```


multicast TpAddress: 10

Please note that the target `TpNode` does not contain a reference to the `TpAddress`. The multicast `TpAddress` is described by a direct reference from the connection.

6.8.11 Diagnostic Connection

A prominent use of the TP in automotive systems is the implementation of diagnostic communication. Data sent from and to the tester frequently exceeds the native size of a communication package on typical bus systems used for this purpose.

However, the mere usage of TP channels for diagnostic purposes is missing one important aspect: TP channels, as defined by the AUTOSAR standard, are unidirectional by nature.

For diagnostic communication, it is very important to be able to define pairs of TP connections that can be taken to send related *request* and *response* messages.

In order to support this use case the meta-class `DiagnosticConnection` has been introduced.

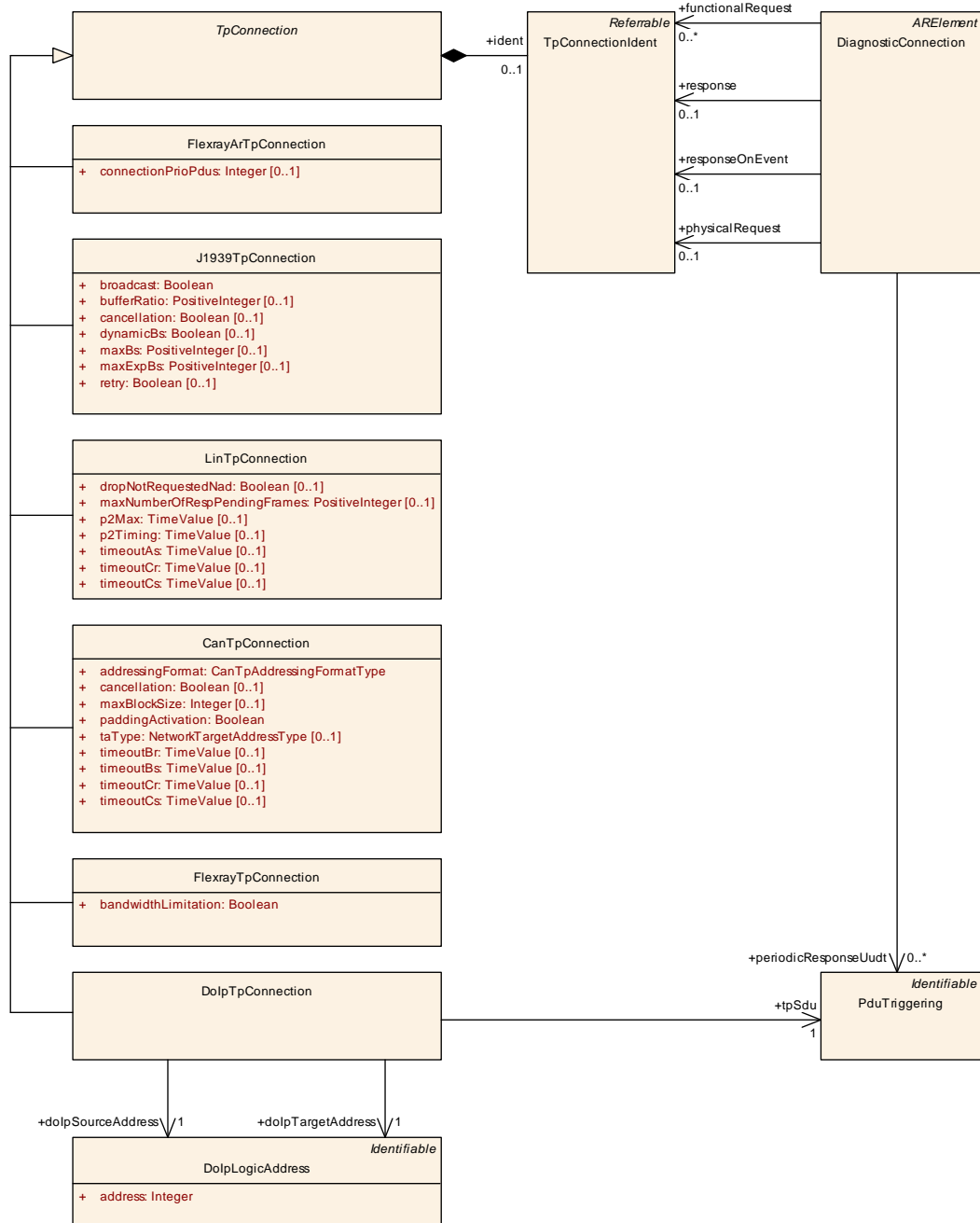


Figure 6.66: Modeling of DiagnosticConnection

[TPS_SYST_05003] Usage of DiagnosticConnection in combination with a TP
 [DiagnosticConnection allows for the dedicated identification of TP connections used for the various diagnostic message sending use cases:

- functionalRequest
- physicalRequest
- responseOnEvent
- response

]()

[TPS_SYST_05004] Usage of [DiagnosticConnection](#) in combination with UUDT [In addition to the usage of TP connections, the [DiagnosticConnection](#) foresees the transmission of UUDT message for **periodic response**. For this purpose, the role [periodicResponseUudt](#) is supported.]()

[constr_1367] [periodicResponseUudt.periodicResponseUudt](#) shall only refer to a [DcmIPdu](#) [If the role [periodicResponseUudt](#) exists then every [PduTriggering](#) referenced in the role [periodicResponseUudt](#) shall only refer to a [DcmIPdu](#).]()

Please note that the meta-class [TpConnectionIdent](#) (derived from [Referrable](#)) has been introduced for the purpose of allowing sub-classes of [TpConnection](#) to become the target of a reference while preserving full backwards-compatibility to the previous modeling.

This means in particular that the existence of a [shortName](#) is only required if the sub-class of [TpConnection](#) shall actually represent the target of a reference in the context of the definition of a [DiagnosticConnection](#).

This, however, is kind of self-evident (because the reference would not work without the existence of a [shortName](#) at the reference target) and therefore it is not necessary to formulate an explicit constraint that clarifies this issue.

[constr_1368] Limitation of the target of references from [DiagnosticConnection](#) [[DiagnosticConnection](#) shall only reference (via the indirection created by [TpConnectionIdent](#)) the following sub-classes of the meta-class [TpConnection](#):

- [CanTpConnection](#)
- [FlexrayTpConnection](#)
- [FlexrayArTpConnection](#)
- [DoIpTpConnection](#)

]()

Class	DiagnosticConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. Tags: atp.recommendedPackage=DiagnosticConnections			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages.





Class	DiagnosticConnection			
periodicResponseUdt	PduTriggering	*	ref	Reference to UUDT responses.
physicalRequest	TpConnectionIdent	0..1	ref	Reference to a physical request message.
response	TpConnectionIdent	0..1	ref	In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed.
responseOnEvent	TpConnectionIdent	0..1	ref	Reference to a ROE message.

Table 6.238: DiagnosticConnection

Class	TpConnection (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	TpConnection Base Class.			
Base	ARObject			
Subclasses	CanTpConnection , DolpTpConnection , EthTpConnection , FlexrayArTpConnection , FlexrayTpConnection , J1939TpConnection , LinTpConnection			
Attribute	Type	Mult.	Kind	Note
ident	TpConnectionIdent	0..1	aggr	This adds the ability to become referable to TpConnection.

Table 6.239: TpConnection

Class	TpConnectionIdent			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referable TpConnection.			
Base	ARObject , Referable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.240: TpConnectionIdent

6.9 Network Management

The NM specification of AUTOSAR consist of a Generic Network Management Interface Module and of bus specific Network management adaptation layers (CanNm, FrNm, UdpNm, J1939Nm). The AUTOSAR Generic NM Interface module acts as a bus-independent adaptation layer between the bus-specific Network Management modules and the AUTOSAR basic software module Communication Manager. The AUTOSAR Generic NM Interface module is represented by `NmCluster`, `NmEcu`, `NmCoordinator` and `NmNode`. The bus-specific Network Management attributes are represented by `BusspecificNmEcu`. See also figure 6.67.

[constr_5032] Maximal one `NmConfig` per `System` is allowed to be defined [Each `System` element is allowed to reference at most one `NmConfig` element with the `fibexElement` reference.]()

[constr_3057] Maximal one `BusspecificNmEcu` per `NmEcu` and bus system is allowed to be defined [For each `NmEcu` at most one `BusspecificNmEcu` per bus system (FlexRay/Can/Udp/J1939) is allowed to be defined.]()

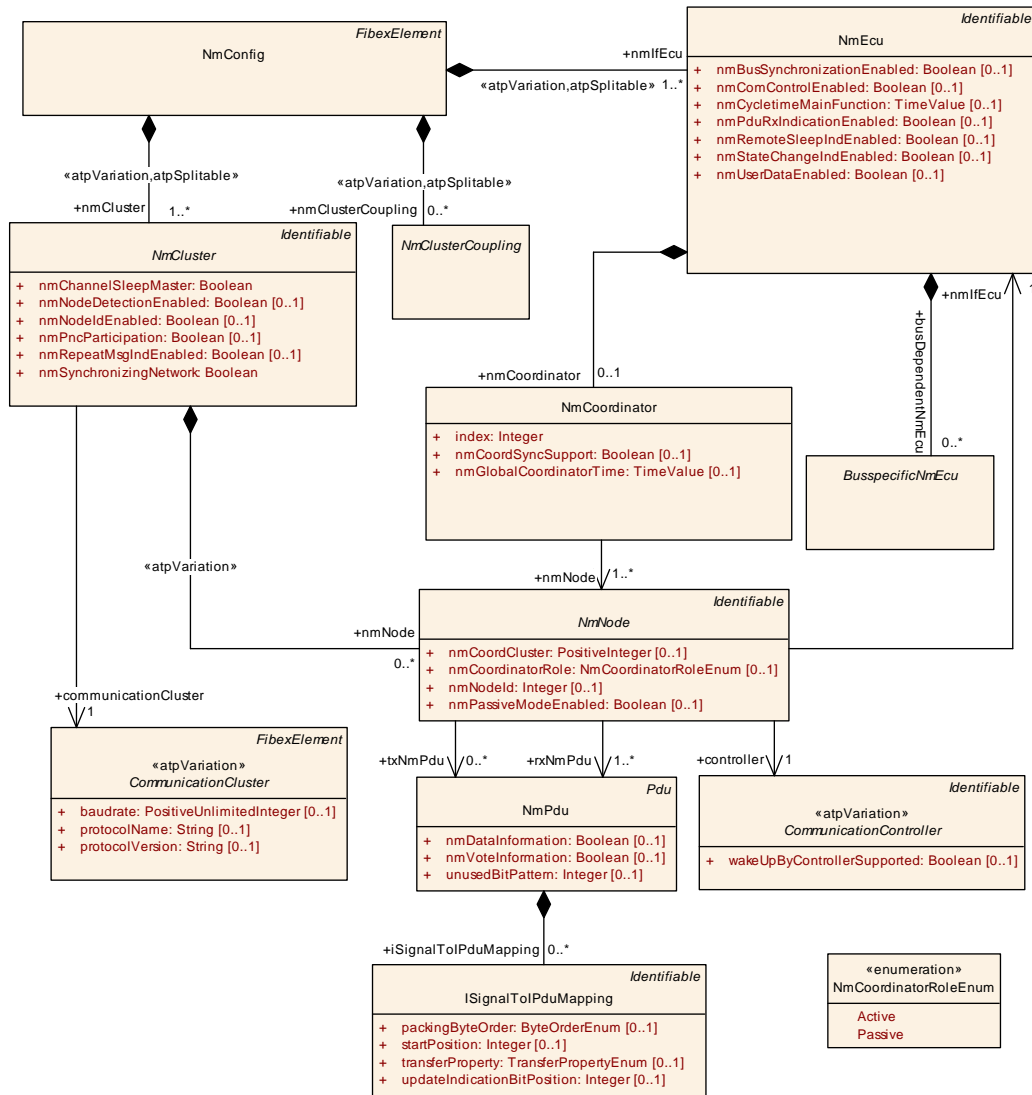


Figure 6.67: Generic Nm elements

The `NmCluster` contains a set of `NmNodes`.

The `NmNodes` are associated with the `CommunicationController` in the topology and belong to exactly one `NmEcu`. The reception and transmission of `NmPdus` is specified with the `rxNmPdu` and `txNmPdu` associations to `NmPdus`.

[TPS_SYST_01107] Definition of `NmCoordinator` [An `nmCoordinator` is connected to two or more `CommunicationClusters` (via `NmNodes`) out of which at least two contain the requirement to shutdown synchronously.] ()

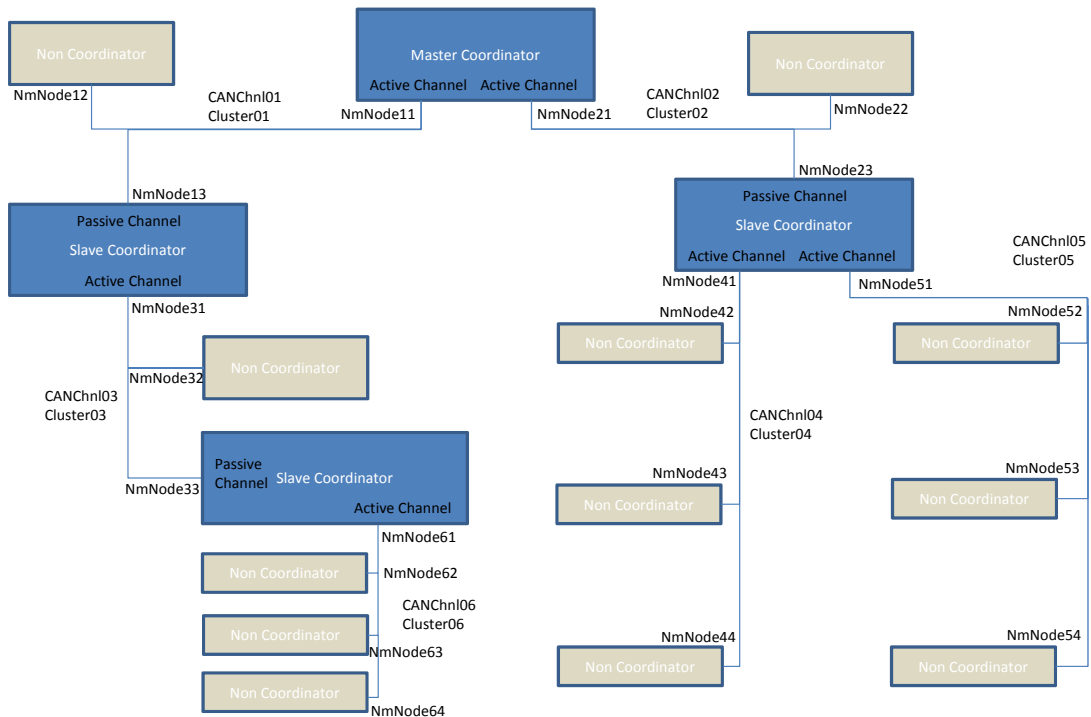


Figure 6.68: Nm Example

Figure 6.68 shows an example and the following section shows how the model shall be used:

NmCluster: Cluster01

- NmNodes:
 - NmNode11 (NmEcu1)
 - NmNode12 (NmEcu2)
 - NmNode13 (NmEcu3)

NmCluster: Cluster02

- NmNodes:
 - NmNode21 (NmEcu1)
 - NmNode22 (NmEcu4)
 - NmNode23 (NmEcu5)

NmCluster: Cluster03

- NmNodes:
 - NmNode31 (NmEcu3)
 - NmNode32 (NmEcu6)

– NmNode33 (NmEcu7)

...

NmEcu1: NmCoordinator (MasterCoordinator)

- NmNode11 (nmCoordinatorRole: Active)
- NmNode21 (nmCoordinatorRole: Active)

NmEcu3: NmCoordinator (SlaveCoordinator)

- NmNode13 (nmCoordinatorRole: Passive)
- NmNode31 (nmCoordinatorRole: Active)

...

Class	NmConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Contains the all configuration elements for AUTOSAR Nm. Tags: atp.ManifestKind=MachineManifest atp.recommendedPackage=NmConfigs			
Base	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
nmCluster	NmCluster	1..*	aggr	Collection of NM Clusters atpVariation: Derived, because cluster can be variable. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName vh.latestBindingTime=postBuild
nmCluster Coupling	NmClusterCoupling	*	aggr	Collection of NmClusterCouplings atpVariation: Derived, because NmCluster can vary. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=NmClusterCoupling, VariationPoint.short Label vh.latestBindingTime=postBuild
nmIfEcu	NmEcu	1..*	aggr	Collection of NM ECUs atpVariation: Derived, because EcuInstance can be variable. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName vh.latestBindingTime=preCompileTime

Table 6.241: NmConfig

Class	NmCluster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Set of NM nodes coordinated with use of the NM algorithm.			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Subclasses	CanNmCluster , FlexrayNmCluster , J1939NmCluster , UdpNmCluster			
Attribute	Type	Mult.	Kind	Note
communication Cluster	CommunicationCluster	1	ref	Association to a CommunicationCluster in the topology description.
nmChannel SleepMaster	Boolean	1	attr	This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision.
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild





Class	NmCluster (abstract)			
nmNodeDetectionEnabled	Boolean	0..1	attr	Enables the Request Repeat Message Request support. Only valid if nmNodeEnabled is set to true.
nmNodeEnabled	Boolean	0..1	attr	Enables the source node identifier.
nmPncParticipation	Boolean	0..1	attr	Defines whether this NmCluster contributes to the partial network mechanism.
nmRepeatMsgIndEnabled	Boolean	0..1	attr	Switch for enabling the Repeat Message Bit Indication.
nmSynchronizingNetwork	Boolean	1	attr	If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.

Table 6.242: NmCluster

[constr_3035] CanNm user data configuration in case NID/CBV are enabled [If NID/CBV are enabled ([nmCbvPosition](#) and [nmNidPosition](#) are configured), there shall not be any user data configured at the position of the respective NID/CBV bytes.]
()

[constr_3044] CBV configuration in case partial network is used [In case a partial network is used the control bit vector (CBV) shall be defined in Byte 0 of the NmPdu ([nmCbvPosition](#) = 0).]()

[constr_3227] NmNode.nmPassiveModeEnabled setting [[NmNode.nmPassiveModeEnabled](#) shall be set to the same value in all NmClusters with the same bus protocol in the scope of one NmEcu.]()

Class	NmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	ECU on which NM is running.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
busDependentNmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes
ecuInstance	EcuInstance	1	ref	Association to an ECUInstance in the topology description.
nmBusSynchronizationEnabled	Boolean	0..1	attr	Enables bus synchronization support.
nmComControlEnabled	Boolean	0..1	attr	Enables the Communication Control support.
nmCoordinator	NmCoordinator	0..1	aggr	Nm ECU may coordinate different clusters.
nmCycletimeMainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the NM Interface in seconds.
nmPduRxIndicationEnabled	Boolean	0..1	attr	Switch for enabling the PDU Rx Indication.





Class	NmEcu			
nmRemoteSleepIndEnabled	Boolean	0..1	attr	Switch for enabling remote sleep indication support.
nmStateChangeIndEnabled	Boolean	0..1	attr	Enables the CAN Network Management state change notification.
nmUserDataEnabled	Boolean	0..1	attr	Switch for enabling user data support.

Table 6.243: NmEcu

Class	BusspecificNmEcu (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Busspecific NmEcu attributes.			
Base	ARObject			
Subclasses	CanNmEcu, FlexrayNmEcu, J1939NmEcu, UdpNmEcu			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.244: BusspecificNmEcu

Class	NmCoordinator			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
index	Integer	1	attr	Identification of the NmCoordinator.
nmCoordSyncSupport	Boolean	0..1	attr	Switch for enabling NmCoordinatorSync (coordination of nested busses) support.
nmGlobalCoordinatorTime	TimeValue	0..1	attr	This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.
nmNode	NmNode	1..*	ref	reference to busses (via NmNodes) that are coordinated by the NmCoordinator.

Table 6.245: NmCoordinator

Class	NmNode (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	CanNmNode, FlexrayNmNode, J1939NmNode, UdpNmNode			
Attribute	Type	Mult.	Kind	Note





Class	NmNode (abstract)			
controller	CommunicationController	1	ref	Association to an CommunicationController in the topology description.
nmCoordCluster	PositiveInteger	0..1	attr	NmCoordinationCluster identification number.
nmCoordinatorRole	NmCoordinatorRoleEnum	0..1	attr	This attribute indicates the role the NM Coordinator will have on this channel.
nmIfEcu	NmEcu	1	ref	Reference to the NmEcu that contains this NmNode. (CommunicationController that is referenced by the NmNode shall be contained in the EcuInstance that is referenced by the NmEcu).
nmNodeId	Integer	0..1	attr	Node identifier of local NmNode. Shall be unique in the NmCluster.
nmPassiveModeEnabled	Boolean	0..1	attr	Enables support of the Passive Mode. The passive mode is configurable per channel.
rxNmPdu	NmPdu	1..*	ref	receive NM Pdu.
txNmPdu	NmPdu	*	ref	transmit NM Pdu

Table 6.246: NmNode

Enumeration	NmCoordinatorRoleEnum
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	Supported NmCoordinator roles.
Literal	Description
Active	Coordinator which "actively" performs NmCoordinator functionality at this channel Tags: atp.EnumerationLiteralIndex=0
Passive	Coordinator which "passively" performs NmCoordinator functionality at this channel - used at NmCoordinatorSync use case. Tags: atp.EnumerationLiteralIndex=1

Table 6.247: NmCoordinatorRoleEnum

Class	NmClusterCoupling (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Attributes that are valid for each of the referenced (coupled) clusters.			
Base	<i>ARObject</i>			
Subclasses	CanNmClusterCoupling , FlexrayNmClusterCoupling , UdpNmClusterCoupling			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.248: NmClusterCoupling

6.9.1 FlexRay Network Management

The following class tables specify the configuration parameters of FlexRay Nm.

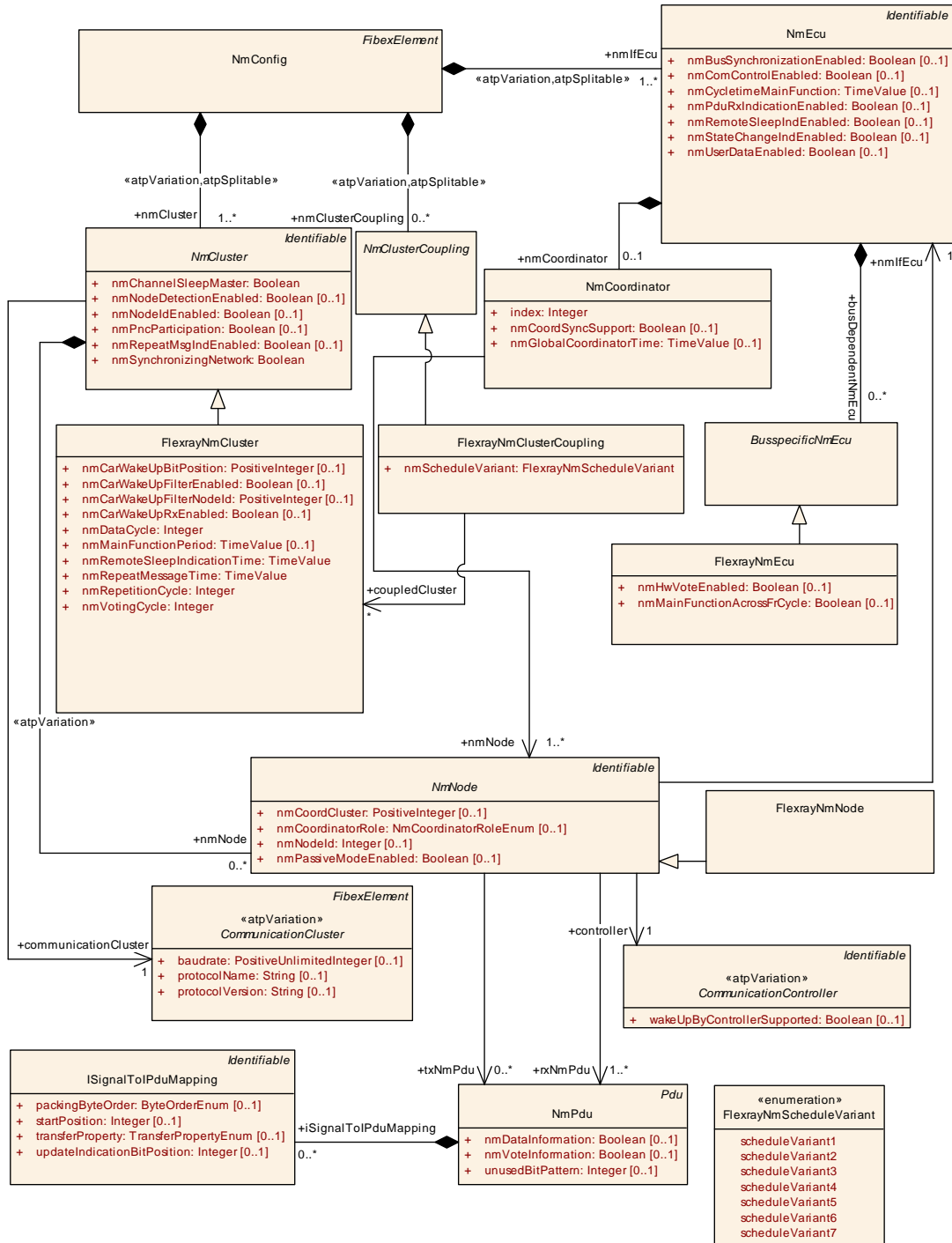


Figure 6.69: FlexRay Network Management Configuration (TransportProtocols: Nm-FlexRayConfiguration)

Class	FlexrayNmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific NM cluster attributes.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>NmCluster</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
nmCarWakeUpBitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUpFilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpFilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If Car WakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpRxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.
nmDataCycle	Integer	1	attr	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.
nmMainFunctionPeriod	TimeValue	0..1	attr	Defines the processing cycle of the main function of FrNm module.
nmRemoteSleepIndicationTime	TimeValue	1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmRepetitionCycle	Integer	1	attr	Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all Flex Ray NmEcus of this FlexRayNmCluster. This value shall be an integral multiple of nmVotingCycle.
nmVotingCycle	Integer	1	attr	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.

Table 6.249: FlexrayNmCluster

Class	FlexrayNmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific attributes.			
Base	<i>ARObject</i> , <i>BusspecificNmEcu</i>			
Attribute	Type	Mult.	Kind	Note
nmHwVoteEnabled	Boolean	0..1	attr	Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.
nmMainFunctionAcrossFrCycle	Boolean	0..1	attr	Parameter describing if the execution of the FrNm_Main function crosses theFlexRay cycle boundary or not.

Table 6.250: FlexrayNmEcu

Class	FlexrayNmClusterCoupling			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay attributes that are valid for each of the referenced (coupled) FlexRay clusters.			
Base	<i>ARObject</i> , <i>NmClusterCoupling</i>			
Attribute	Type	Mult.	Kind	Note
coupledCluster	FlexrayNmCluster	*	ref	Reference to coupled FlexRay Clusters.
nmSchedule Variant	FlexrayNmSchedule Variant	1	attr	FrNm schedule variant according to FrNm SWS.

Table 6.251: FlexrayNmClusterCoupling

Class	FlexrayNmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific NM Node attributes.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>NmNode</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.252: FlexrayNmNode

Enumeration	FlexrayNmScheduleVariant
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	FrNm schedule variant according to FrNm SWS.
Literal	Description
scheduleVariant1	NM-Vote and NM Data transmitted within one PDU in static segment. The NM-Vote has to be realized as separate bit within the PDU. Tags: atp.EnumerationLiteralIndex=0
scheduleVariant2	NM-Vote and NM-Data transmitted within one PDU in dynamic segment. The presence (or non-presence) of the PDU corresponds to the NM-Vote Tags: atp.EnumerationLiteralIndex=1
scheduleVariant3	NM-Vote and NM-Data are transmitted in the static segment in separate PDUs. This alternative is not recommended => Alternative 1 should be used instead. Tags: atp.EnumerationLiteralIndex=2
scheduleVariant4	NM-Vote transmitted in static and NM-Data transmitted in dynamic segment. Tags: atp.EnumerationLiteralIndex=3
scheduleVariant5	NM-Vote is transmitted in dynamic and NM-Data is transmitted in static segment. This alternative is not recommended => Variants 2 or 6 should be used instead. Tags: atp.EnumerationLiteralIndex=4
scheduleVariant6	NM-Vote and NM-Data are transmitted in dynamic segment in separate PDUs. Tags: atp.EnumerationLiteralIndex=5
scheduleVariant7	NM-Vote and a copy of the CBV are transmitted in the static segment (using the FlexRay NM Vector support) and NM-Data is transmitted in the dynamic segment Tags: atp.EnumerationLiteralIndex=6

Table 6.253: FlexrayNmScheduleVariant

6.9.2 CAN Network Management

The following class tables specify the configuration parameters of CAN Nm.

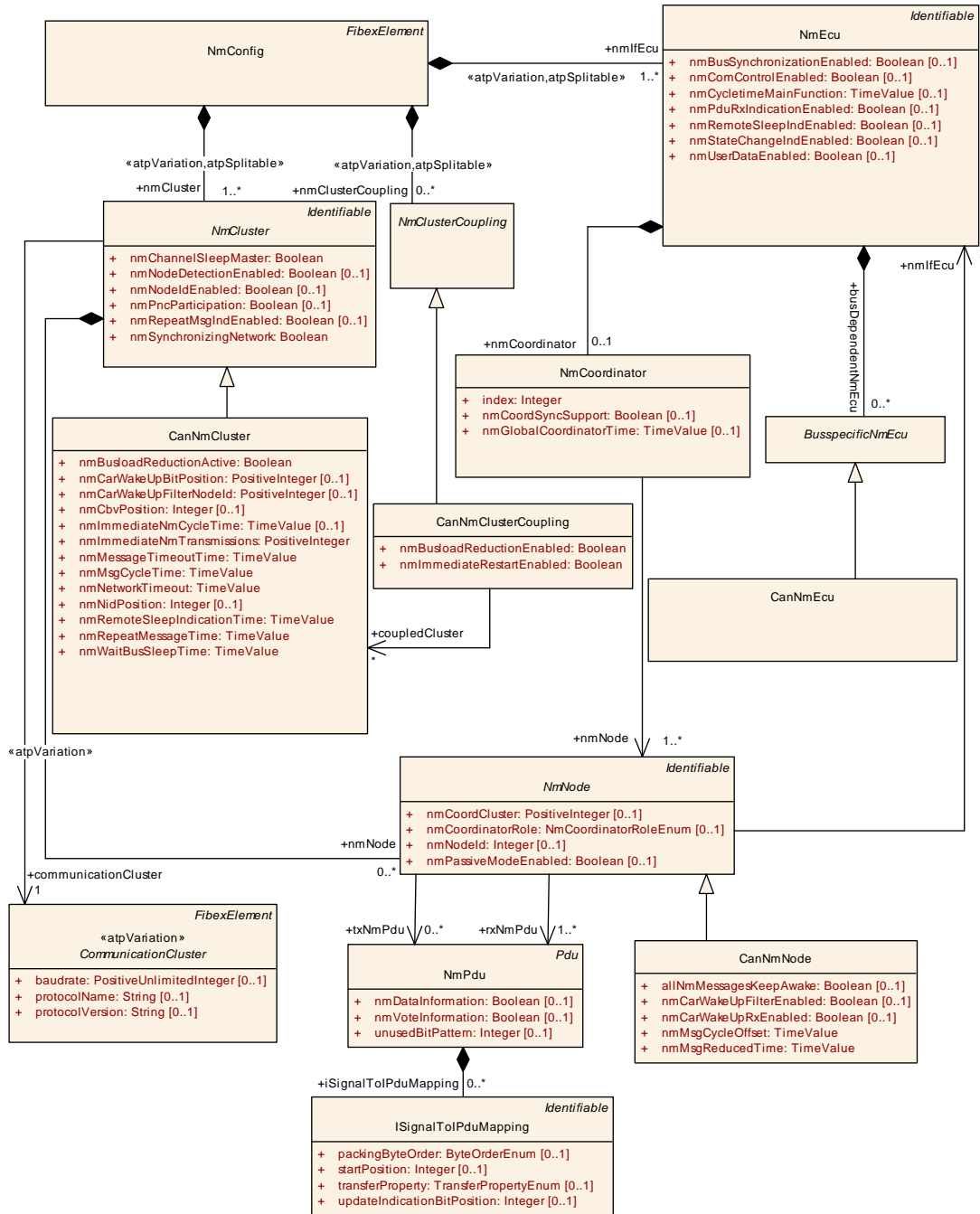


Figure 6.70: CAN Network Management Configuration (TransportProtocols: NmCanConfiguration)

Class		CanNmCluster		
Package		M2::AUTOSARTemplates::SystemTemplate::NetworkManagement		
Note		Can specific NmCluster attributes		
Base		ARObject, Identifiable , MultilanguageReferrable , NmCluster , Referrable		
Attribute	Type	Mult.	Kind	Note
nmBusloadReductionActive	Boolean	1	attr	It determines if bus load reduction for the respective Can Nm channel is active or not.
nmCarWakeUpBitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUpFilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering.
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
nmImmediateNmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This parameter is only valid if CanNmImmediateNmTransmissions is greater one.
nmImmediateNmTransmissions	PositiveInteger	1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessageTimeoutTime	TimeValue	1	attr	Timeout of an NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
nmMsgCycleTime	TimeValue	1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetworkTimeout	TimeValue	1	attr	Network Timeout for NmPdus in seconds It denotes the time how long the CanNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmRemoteSleepIndicationTime	TimeValue	1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBusSleepTime	TimeValue	1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.

Table 6.254: CanNmCluster

[constr_3069] Allowed [CanNmCluster.nmNidPosition](#) values [The value of [CanNmCluster.nmNidPosition](#) shall only be set to either 0 or 1.]()

[constr_3070] Allowed [CanNmCluster.nmCbvPosition](#) values [The value of [CanNmCluster.nmCbvPosition](#) shall only be set to either 0 or 1.]()

[constr_3071] **CanNmCluster.nmCbvPosition** and **CanNmCluster.nmNidPosition** shall never have the same value [CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value.]()

Class	CanNmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN specific attributes.			
Base	ARObject, BusspecificNmEcu			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.255: CanNmEcu

Class	CanNmClusterCoupling			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN attributes that are valid for each of the referenced (coupled) CAN clusters.			
Base	ARObject, NmClusterCoupling			
Attribute	Type	Mult.	Kind	Note
coupledCluster	CanNmCluster	*	ref	Reference to coupled CAN Clusters.
nmBusloadReductionEnabled	Boolean	1	attr	Enables busload reduction support
nmImmediateRestartEnabled	Boolean	1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.256: CanNmClusterCoupling

Class	CanNmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN specific NM Node attributes.			
Base	ARObject, Identifiable , MultilanguageReferrable , NmNode , Referrable			
Attribute	Type	Mult.	Kind	Note
allNmMessagesKeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake
nmCarWakeUpFilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported.
nmCarWakeUpRxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.
nmMsgCycleOffset	TimeValue	1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.





<i>Class</i>	CanNmNode			
nmMsg ReducedTime	TimeValue	1	attr	Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.

Table 6.257: CanNmNode

6.9.3 UDP Network Management

The UPD Nm model is similar to the Nm models of the other communication buses but there are some specific characteristics due to the modeling of `EthernetCluster` and `EthernetPhysicalChannel` (see also chapter 3.3.6).

The `UdpNmCluster` corresponds to one `EthernetPhysicalChannel` (VLAN). Therefore it is required that for each `EthernetPhysicalChannel` on one `EthernetCluster` a respective `UdpNmCluster` with a reference to the `EthernetPhysicalChannel` is created. All of these `UdpNmClusters` point to the same `EthernetCluster` which the `EthernetPhysicalChannels` are contained in.

Thus, additionally to the reference from `NmCluster` to the `CommunicationCluster` (which applies to all Nm models), there is need for an Ethernet specific reference from the `UdpNmCluster` to the `EthernetPhysicalChannel`. This allows to specify for which VLAN this `UdpNmCluster` applies.

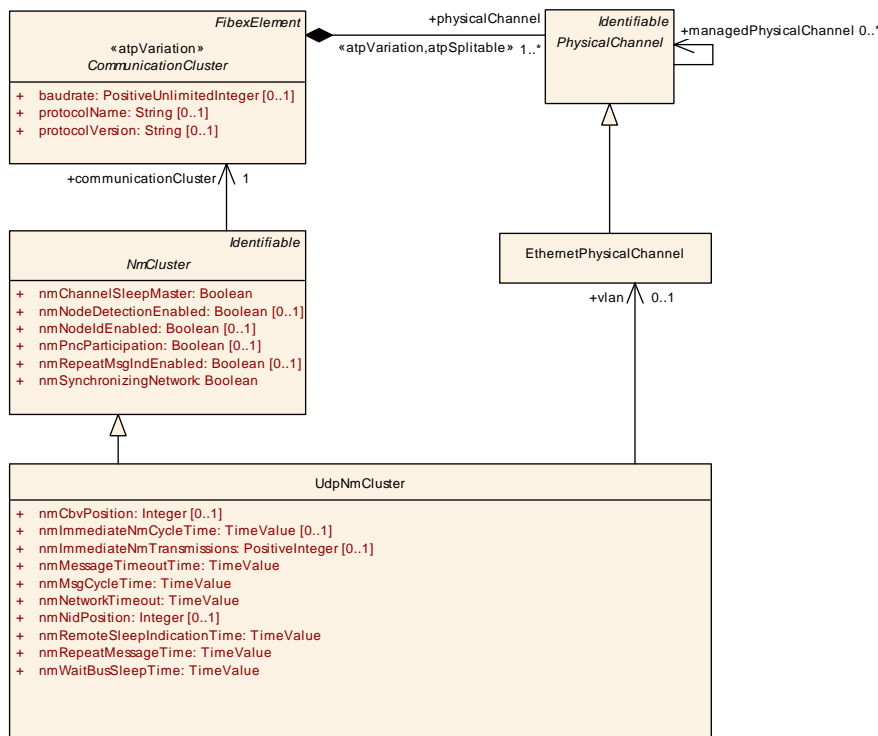


Figure 6.71: UdpNmCluster structure

The following class tables specify the configuration parameters of UDP Nm.

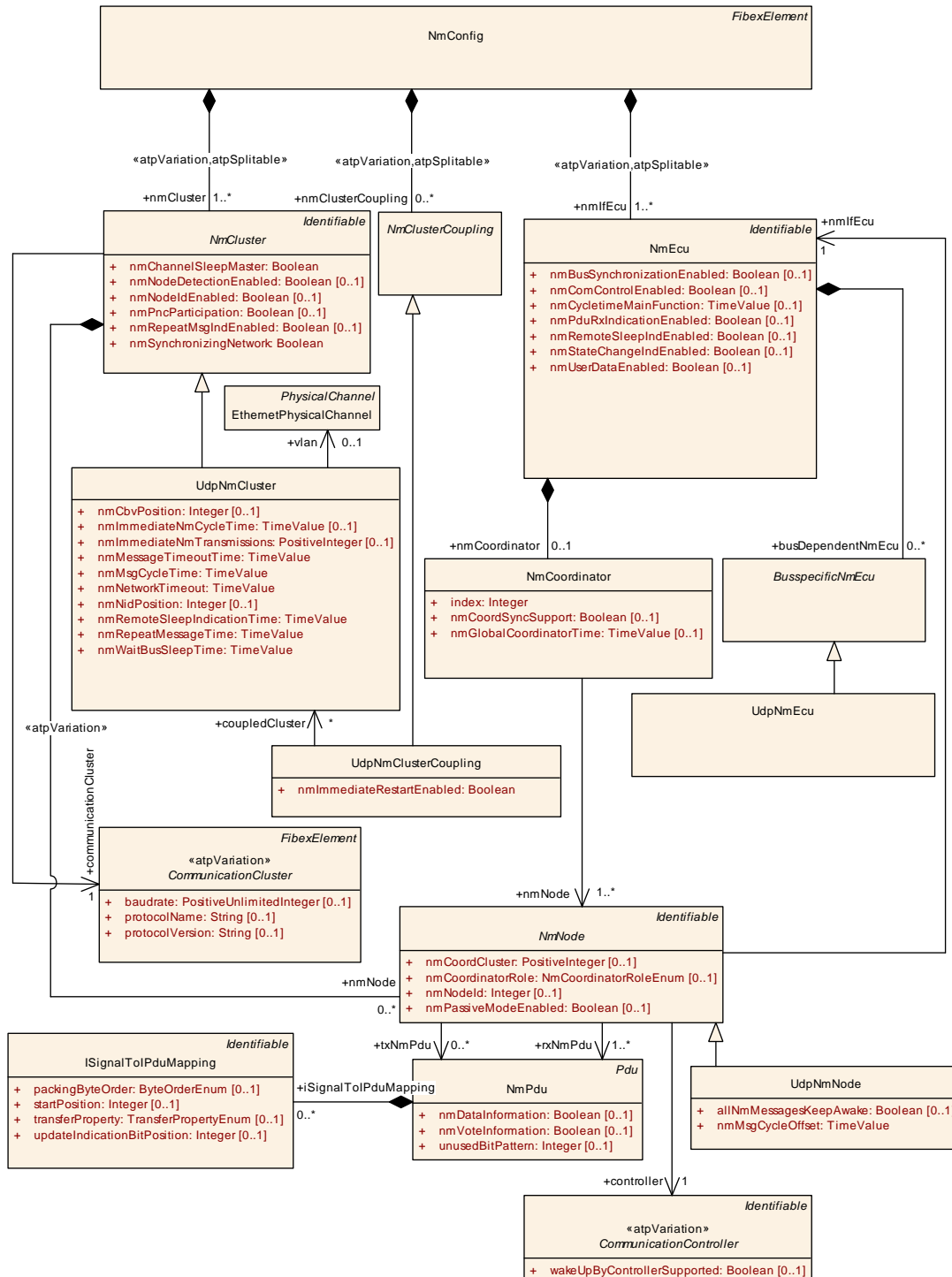


Figure 6.72: UDP Network Management Configuration (TransportProtocols: NmUdpConfiguration)

Class	UdpNmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NmCluster attributes Tags: atp.ManifestKind=MachineManifest			
Base	ARObject, Identifiable , MultilanguageReferrable , NmCluster , Referrable			
Attribute	Type	Mult.	Kind	Note
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position).
nmImmediateNmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmImmediateNmTransmissions is greater one.
nmImmediateNmTransmissions	PositiveInteger	0..1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessageTimeoutTime	TimeValue	1	attr	Timeout of a NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
nmMsgCycleTime	TimeValue	1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetworkTimeout	TimeValue	1	attr	Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu.
nmRemoteSleepIndicationTime	TimeValue	1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBusSleepTime	TimeValue	1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.
vlan	EthernetPhysicalChannel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to.

Table 6.258: UdpNmCluster

[constr_3078] Allowed [UdpNmCluster.nmNidPosition](#) values [The value of [UdpNmCluster.nmNidPosition](#) shall only be set to either 0 or 1.]()

[constr_3079] Allowed [UdpNmCluster.nmCbvPosition](#) values [The value of [UdpNmCluster.nmCbvPosition](#) shall only be set to either 0 or 1.]()

[constr_3080] [UdpNmCluster.nmCbvPosition](#) and [UdpNmCluster.nmNidPosition](#) shall never have the same value [[UdpNmCluster.nmCbvPosition](#) and [UdpNmCluster.nmNidPosition](#) shall never have the same value.]()

Class	UdpNmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp NM specific ECU attributes.			
Base	ARObject, BusspecificNmEcu			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.259: UdpNmEcu

Class	UdpNmClusterCoupling			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp attributes that are valid for each of the referenced (coupled) UdpNm clusters.			
Base	ARObject, NmClusterCoupling			
Attribute	Type	Mult.	Kind	Note
coupledCluster	UdpNmCluster	*	ref	Reference to coupled UdpNm Clusters.
nmImmediateRestartEnabled	Boolean	1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.260: UdpNmClusterCoupling

Class	UdpNmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NM Node attributes. Tags: atp.ManifestKind=MachineManifest			
Base	ARObject, Identifiable , MultilanguageReferrable , NmNode , Referrable			
Attribute	Type	Mult.	Kind	Note
allNmMessagesKeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake
nmMsgCycleOffset	TimeValue	1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.

Table 6.261: UdpNmNode

6.9.4 J1939 Network Management

The following class tables specify the configuration parameters of J1939 Nm.

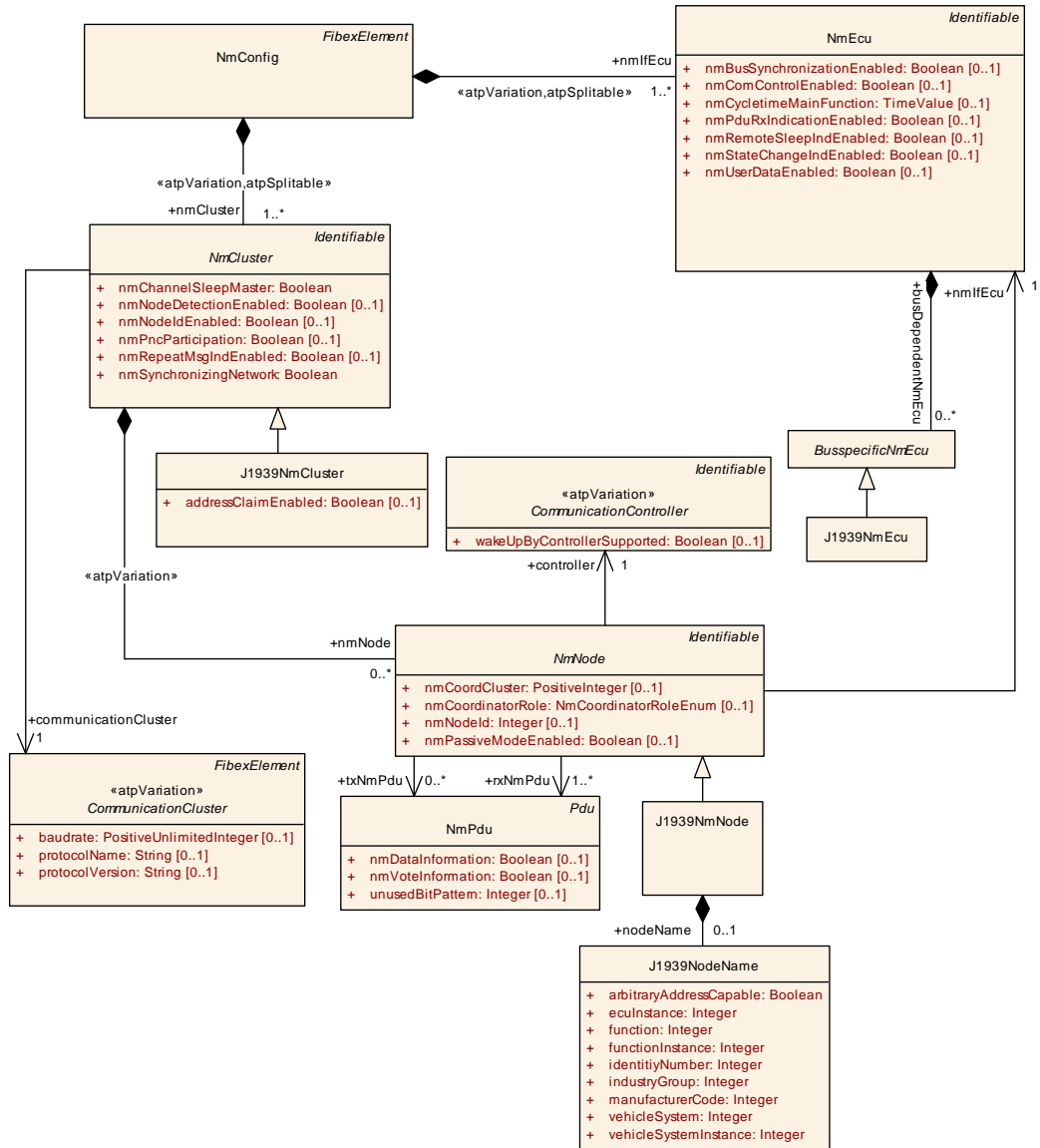


Figure 6.73: J1939 Network Management Configuration (TransportProtocols: NmJ1939Configuration)

Class	J1939NmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NmCluster attributes			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>NmCluster</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
addressClaim Enabled	Boolean	0..1	attr	This attribute specifies whether the J1939Nm Bsw module is used or not. If this attribute is set to false then the J1939Nm configuration shall not be derived from the system description. But even in this case the nmNode might still be necessary for the J1939Rm and J1939Tp.

Table 6.262: J1939NmCluster

Class	J1939NmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NM Node attributes.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>NmNode</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
nodeName	J1939NodeName	0..1	aggr	nodeName configuration

Table 6.263: J1939NmNode

Class	J1939NodeName			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	This element contains attributes to configure the J1939NmNode NAME.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
arbitrary Address Capable	Boolean	1	attr	Arbitrary Address Capable field of the NAME of this node.
ecuInstance	Integer	1	attr	ECU Instance field of the NAME of this node.
function	Integer	1	attr	Function field of the NAME of this node.
function Instance	Integer	1	attr	Function Instance field of the NAME of this node.
identityNumber	Integer	1	attr	Identity Number field of the NAME of this node.
industryGroup	Integer	1	attr	Industry Group field of the NAME of this node.
manufacturer Code	Integer	1	attr	Manufacturer Code field of the NAME of this node.
vehicleSystem	Integer	1	attr	Vehicle System field of the NAME of this node.
vehicleSystem Instance	Integer	1	attr	Vehicle System Instance field of the NAME of this node.

Table 6.264: J1939NodeName

[constr_3102] Restriction on usage of [J1939NodeName](#) attributes [A [J1939NmCluster](#) shall not aggregate two [J1939NmNodes](#) with identical [J1939NodeName](#) attributes.]()

[constr_5029] J1939NmCluster is not allowed to reference a TtcanCluster [A [J1939NmCluster](#) is not allowed to reference a [TtcanCluster](#) in the role [communicationCluster](#).]()

[constr_3103] Range of ecuInstance [The allowed values of [ecuInstance](#) range from 0 to 7.]()

[constr_3104] Range of function [The allowed values of [function](#) range from 0 to 255.]()

[constr_3105] Range of functionInstance [The allowed values of [functionInstance](#) range from 0 to 31.]()

[constr_3106] Range of identityNumber [The allowed values of [identityNumber](#) range from 0 to 2097151.]()

[constr_3107] Range of industryGroup [The allowed values of [industryGroup](#) range from 0 to 7.]()

[constr_3108] Range of manufacturerCode [The allowed values of [manufacturerCode](#) range from 0 to 2047.]()

[constr_3109] Range of vehicleSystem [The allowed values of [vehicleSystem](#) range from 0 to 127.]()

[constr_3110] Range of vehicleSystemInstance [The allowed values of [vehicleSystemInstance](#) range from 0 to 15.]()

Class	J1939NmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 NmEcu specific attributes.			
Base	ARObject , BusspecificNmEcu			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.265: J1939NmEcu

6.9.4.1 J1939SharedAddressCluster

There are two ways of identifying source and target nodes in routing relations in J1939 networks (see [[TPS_SYST_02107](#)] and [[TPS_SYST_02108](#)]).

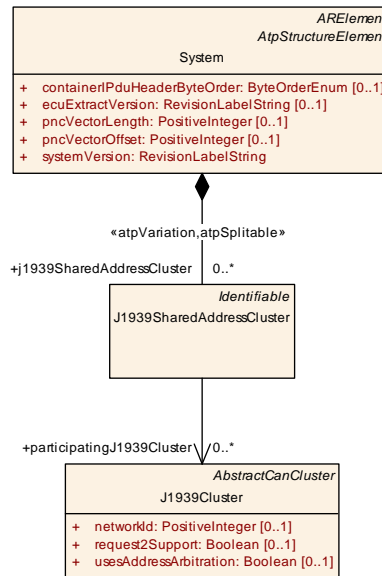


Figure 6.74: J1939SharedAddressCluster

Class	J1939SharedAddressCluster			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	This meta-class represents the ability to identify several J1939Clusters that share a common address space for the routing of messages			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
participating J1939Cluster	J1939Cluster	*	ref	This identifies the J1939Clusters that share a common address space

Table 6.266: J1939SharedAddressCluster

[TPS_SYST_02107] Shared address space for J1939 routing relations [Address claims are routed between several CommunicationClusters independent of whether there are actual routings between individual nodes on respective CommunicationClusters. This means that the overall number of nodes in the shared CommunicationCluster cannot exceed 254, independently of the routing relations.] (RS_SYST_00038)

[TPS_SYST_02108] Address proxying for J1939 routing relations [The gateway claims all addresses used in routed messages on those CommunicationClusters to which the actual nodes are not connected. Thereby the address spaces are separate and only the nodes participating in a routing appear on more than one CommunicationCluster. The total number of nodes in the participating CommunicationClusters can be higher than 254, and the address arbitration is faster with less conflicts.] (RS_SYST_00038)

[TPS_SYST_02109] Absence of participatingJ1939Cluster to a J1939Cluster [If J1939Clusters exist that participate in a routing relation but are not referenced in the role J1939SharedAddressCluster.participatingJ1939Cluster by the same J1939SharedAddressCluster then gateway

shall apply the address proxying according to [TPS_SYST_02108].](RS_SYST_00038)

6.9.5 Managed Channels

There is the use case to transmit NM frames on one VLAN (`EthernetPhysicalChannel`) and the application data on different VLANs. At the same time it shall be possible to indicate that a VLAN uses Network Management although no `NmPdus` are defined on this channel.

A reference between `PhysicalChannels` is used to express such a setting: The managing `PhysicalChannel` that contains configured `NmPdus` references `PhysicalChannels` in the role `managedPhysicalChannel`.

Since the reference `managedPhysicalChannel` is available on the abstract `PhysicalChannel` element it is not only usable in case of `UdpNm` and VLANs but also for similar cases on other bus networks.

[constr_3479] `PhysicalChannel` is not allowed to be a `managedPhysicalChannel` and a managing `PhysicalChannel` [If a `PhysicalChannel` is referenced in role `managedPhysicalChannel`, then it shall not be the source of another `managedPhysicalChannel` relation.]()

[constr_3480] `PhysicalChannel` shall be referenced in the role `managedPhysicalChannel` only once [A `PhysicalChannel` shall be referenced in the role `managedPhysicalChannel` only up to once.]()

[constr_3481] `UdpNmCluster` is not allowed to reference a `managedPhysicalChannel` in the role `vlan` [If an `EthernetPhysicalChannel` is target of a `managedPhysicalChannel` reference, then no `UdpNmCluster` shall reference this `managedPhysicalChannel` in the role `vlan`.]()

[constr_3482] `NmCluster` is not allowed to reference a `CommunicationCluster` that aggregates a `managedPhysicalChannel` [If a `PhysicalChannel`, except `EthernetPhysicalChannel`, is target of a `managedPhysicalChannel`, then the aggregating `CommunicationCluster` shall not be referenced by any `NmCluster` in the role `communicationCluster`.]()

[constr_3483] The same `PhysicalChannel` is not allowed to be the source and the target of `managedPhysicalChannel` references [If a `PhysicalChannel` is source of any `managedPhysicalChannel` reference, then it shall not be the target of any `managedPhysicalChannel` relation.]()

6.10 Bus Mirroring

Many communication buses in a vehicle are not directly accessible by a tester. To allow a tester to listen to the traffic on such internal communication buses the bus mirroring

is introduced. The bus mirroring collects traffic from such an internal communication bus and forwards it to an intermediate destination bus or to a destination bus that is accessible by the tester.

Testers connected via CAN will receive unmodified CAN frames and LIN frames with special CAN IDs. Testers connected via Ethernet will receive a stream containing current time, identification, and content of CAN, LIN, and FlexRay frames.

On intermediate FlexRay buses, a set of PDUs is used to transport streams of mirrored frames with the same layout as on Ethernet.

[TPS_SYST_02202] Modeling of bus mirroring [The [BusMirrorChannelMapping](#) defines the bus mirroring in which the communication traffic of the `sourceChannel` is forwarded to the `targetChannel`.] ()

Class	BusMirrorChannelMapping (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines a bus mirroring in which the traffic from one communication bus (<code>sourceChannel</code>) is forwarded to another one (<code>targetChannel</code>).			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	BusMirrorChannelMappingCan , BusMirrorChannelMappingFlexray , BusMirrorChannelMappingIp , BusMirrorChannelMappingUserDefined			
Attribute	Type	Mult.	Kind	Note
<code>sourceChannel</code>	BusMirrorChannel	0..1	aggr	Defines the <code>sourceChannel</code> from which frames are received.
<code>targetChannel</code>	BusMirrorChannel	0..1	aggr	Defines the <code>targetChannel</code> to which frames are forwarded.
<code>targetPduTriggering</code>	PduTriggering	*	ref	Reference to the <code>PduTriggering</code> that is used for transmission of the mirrored frames on the <code>targetChannel</code> . Please note that on FlexRay several <code>targetPduTriggerings</code> may be used. For all other communication channels only a single <code>targetPduTriggering</code> is supported. Stereotypes: <code>atpVariation</code> Tags: <code>vh.latestBindingTime=postBuild</code>

Table 6.267: BusMirrorChannelMapping

[constr_3464] Allowed Pdu type on [BusMirrorChannelMapping.targetChannel](#) [Each `PduTriggering` that is referenced by [BusMirrorChannelMapping](#) in the role `targetPduTriggering` is only allowed to reference a `GeneralPurposeIPdu` of category `BUS_MIRRORING`.] ()

Class	BusMirrorChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element assigns a <code>busMirrorNetworkId</code> to the referenced channel.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note





Class	BusMirrorChannel			
busMirrorNetworkId	PositiveInteger	1	attr	This attribute defines the networkId of the communication channel.
channel	PhysicalChannel	0..1	ref	Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime

Table 6.268: BusMirrorChannel

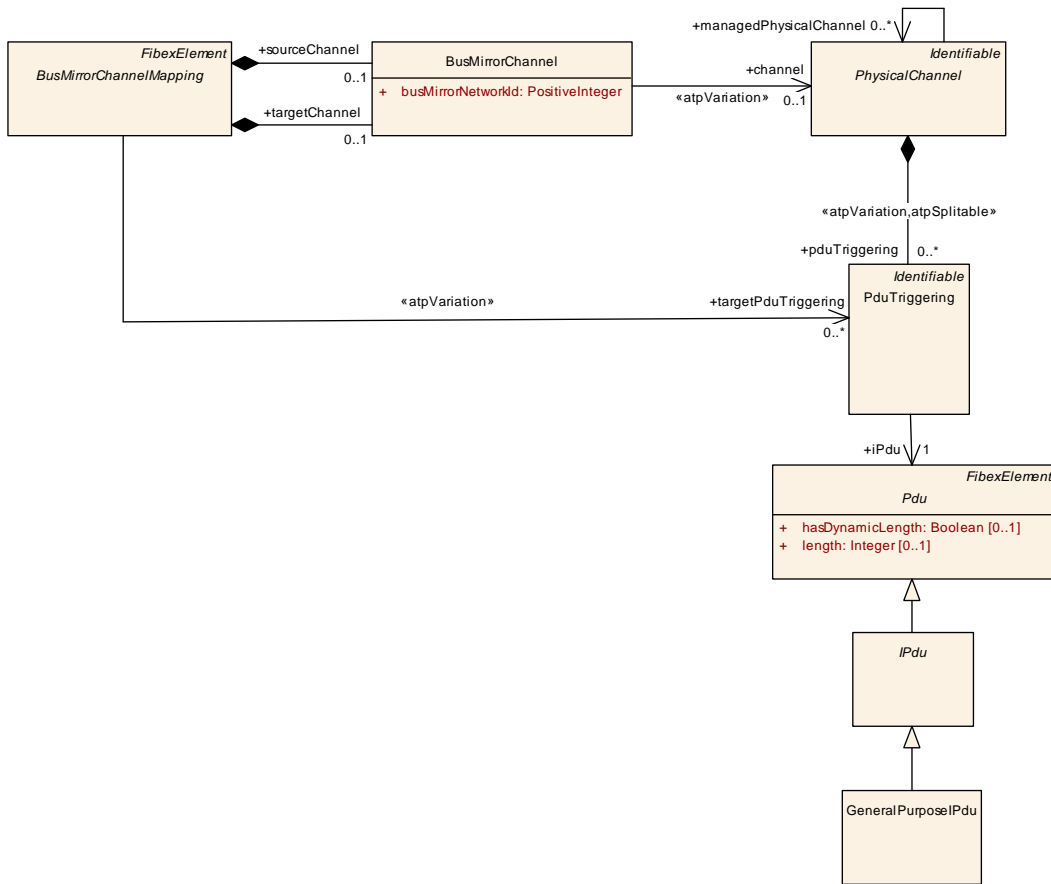


Figure 6.75: Bus mirroring

[constr_3465] Identical `BusMirrorChannel.busMirrorNetworkId` for `BusMirrorChannels` referencing the same `PhysicalChannel` [The attribute `BusMirrorChannel.busMirrorNetworkId` shall be identical in all `BusMirrorChannels` that are referencing the same `PhysicalChannel` in the scope of the `System`.] ()

[constr_3466] Unique `BusMirrorChannel.busMirrorNetworkIds` for each specialization of `PhysicalChannel` [The attribute `BusMirrorChannel.busMirrorNetworkId` associated with `PhysicalChannels` that have the same specialization (e.g. all `CanPhysicalChannels`) shall have unique `BusMirrorChannel.busMirrorNetworkIds` within the scope of the `System`.] ()

6.10.1 CAN Destination Channel

[TPS_SYST_02203] BusMirroring to CAN destination channel [In case of CAN to CAN and LIN to CAN the [BusMirrorChannelMappingCan](#) meta-class shall be used for the modeling of the bus mirroring.] ()

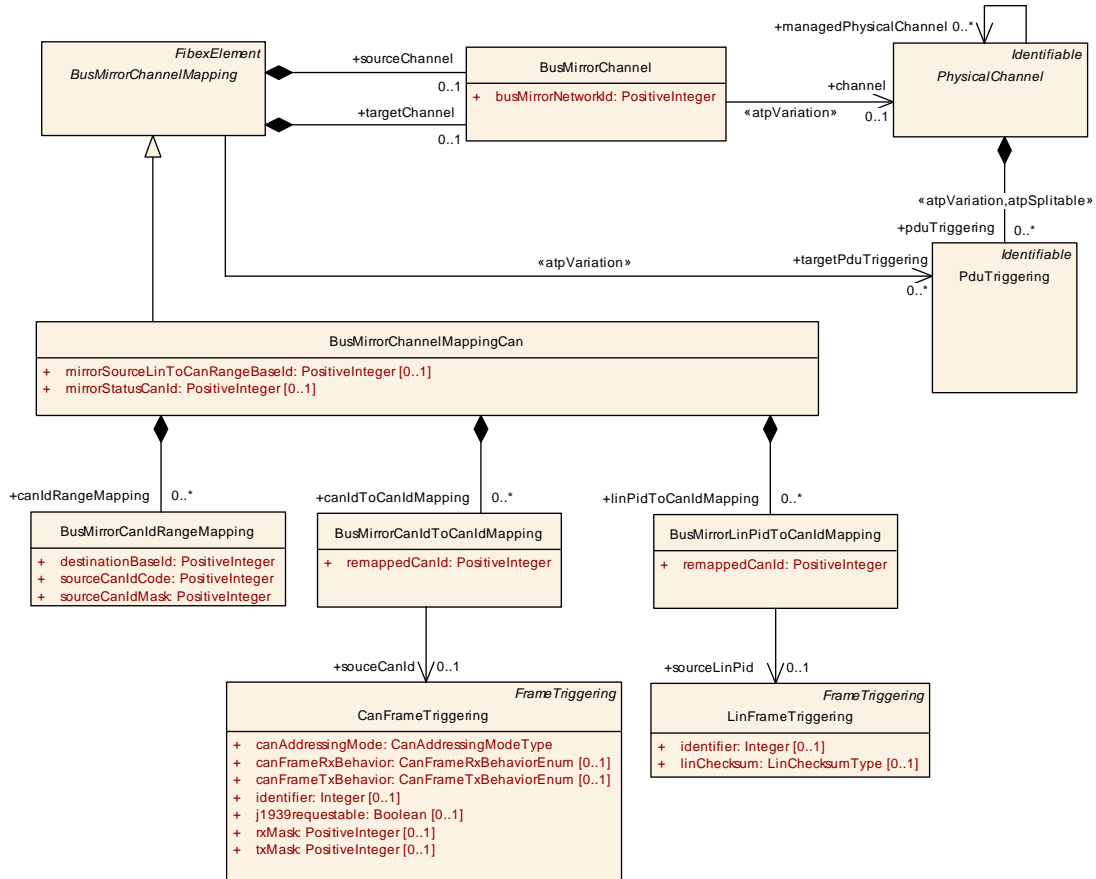


Figure 6.76: Bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel

Class	BusMirrorChannelMappingCan			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel. Tags: atp.recommendedPackage=BusMirrorChannelMappings			
Base	ARObject, BusMirrorChannelMapping , CollectableElement , FibexElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
canIdRangeMapping	BusMirrorCanIdRangeMapping	*	aggr	Rules for remapping of a set of CAN IDs.
canIdToCanIdMapping	BusMirrorCanIdToCanIdMapping	*	aggr	Rules for remapping of single CanIds.





Class	BusMirrorChannelMappingCan			
linPidToCanId Mapping	BusMirrorLinPidToCanIdMapping	*	aggr	Rules for remapping of single LIN Frames.
mirrorSourceLinToCanRange BaseId	PositiveInteger	0..1	attr	Base ID merged with the LIN frame ID to form the CAN ID. Only required when a BusMirrorChannel that refers to a LinPhysicalChannel in the role channel is referenced in the role sourceChannel.
mirrorStatus CanId	PositiveInteger	0..1	attr	CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.

Table 6.269: BusMirrorChannelMappingCan

[constr_3467] CanPhysicalChannel as destination channel of BusMirrorChannelMappingCan [The [BusMirrorChannel](#) that is aggregated by [BusMirrorChannelMappingCan](#) shall only reference a [CanPhysicalChannel](#) in the role [targetChannel](#).]()

[constr_3468] BusMirrorChannelMappingCan.targetPduTriggering restriction [[BusMirrorChannelMappingCan](#) is allowed to reference only one single [PduTriggering](#) in the role [targetPduTriggering](#).]()

[constr_3469] CanFrameTriggering.txMask setting for the destination frame [The [CanFrameTriggering](#) of a [Frame](#) that contains a [Pdu](#) of which the [PduTriggering](#) is referenced by [BusMirrorChannelMappingCan](#) in the role [targetPduTriggering](#) shall set the [txMask](#) to 0.]()

[constr_5051] Existence of CanFrameTriggering.identifier in case of bus mirror target [The [CanFrameTriggering](#) of a [Frame](#) that contains a [Pdu](#) of which the [PduTriggering](#) is referenced by [BusMirrorChannelMappingCan](#) in the role [targetPduTriggering](#) shall not define an [identifier](#).]()

[constr_3470] PaddingValue used to transmit the Pdu on a Can-Fd destination bus [In case that the [BusMirrorChannelMappingCan](#) references a [PduTriggering](#) in the role [targetPduTriggering](#) and

- the [CanFrameTriggering](#) of the [Frame](#) that contains this [targetPduTriggering](#) has the [canFrameTxBehavior](#) set to [canFd](#) and
- the [CanFrameTriggering](#) has a reference to an “out” [FramePort](#) (i.e. the [Frame](#) is transmitted by an [Ecu](#) on a [Can-Fd](#) destination bus) and
- the [CommunicationController](#) of the transmitting [EcuInstance](#) that is referenced via the [CommunicationConnector](#) by the [PhysicalChannel](#) on which the [targetPduTriggering](#) is located then the [CanControllerFdConfiguration.paddingValue](#) or [CanControllerFdConfigurationRequirements.paddingValue](#) shall have the value 0.

]()

Class	BusMirrorCanIdRangeMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines a rule for remapping a set of CAN IDs.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
destinationBaseId	PositiveInteger	1	attr	Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.
sourceCanIdCode	PositiveInteger	1	attr	Value to match masked original CAN IDs.
sourceCanIdMask	PositiveInteger	1	attr	Mask applied to original CAN IDs before comparison.

Table 6.270: BusMirrorCanIdRangeMapping

Class	BusMirrorCanIdToCanIdMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines a rule for remapping a single CAN ID.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
remappedCanId	PositiveInteger	1	attr	This attribute defines the CanId on the targetChannel.
sourceCanId	CanFrameTriggering	0..1	ref	This reference points to the sourceFrame with sourceCanId on the sourceChannel.

Table 6.271: BusMirrorCanIdToCanIdMapping

Class	BusMirrorLinPidToCanIdMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines a rule for remapping a single LIN Frame.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
remappedCanId	PositiveInteger	1	attr	This attribute defines the CanId on the targetChannel.
sourceLinPid	LinFrameTriggering	0..1	ref	This reference points to the sourceFrame with sourceCanId on the sourceChannel.

Table 6.272: BusMirrorLinPidToCanIdMapping

6.10.2 FlexRay Destination Channel

[TPS_SYST_02204] **BusMirroring to FlexRay destination channel** [In case of CAN to FlexRay, LIN to FlexRay and FlexRay to FlexRay the [BusMirrorChannelMappingFlexray](#) meta-class shall be used for the modeling of the bus mirroring.]()

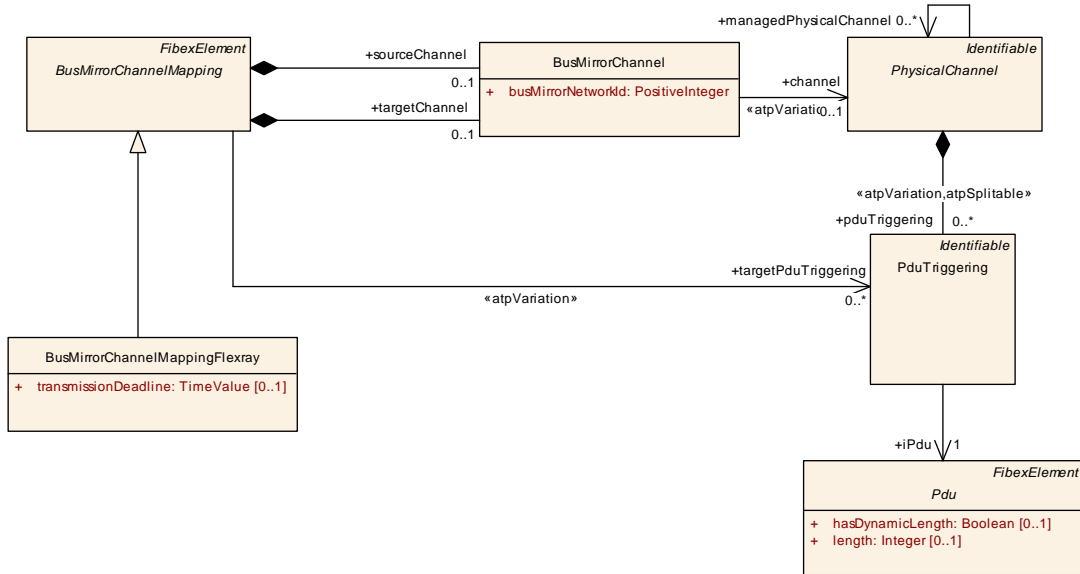


Figure 6.77: Bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel

Class	BusMirrorChannelMappingFlexray			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel. Tags: atp.recommendedPackage=BusMirrorChannelMappings			
Base	ARObject, BusMirrorChannelMapping, CollectableElement, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
transmissionDeadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table 6.273: BusMirrorChannelMappingFlexray

[constr_3471] FlexrayPhysicalChannel as destination channel of BusMirrorChannelMappingFlexray [The BusMirrorChannel that is aggregated by BusMirrorChannelMappingFlexray shall only reference a FlexrayPhysicalChannel in the role targetChannel.]()

[constr_3472] Number of BusMirrorChannels derived for one FlexrayCluster [For each FlexrayCluster, only one BusMirrorChannel shall be derived. I.e. if both channels A and B are derived, only one of the two FlexrayPhysicalChannels of one FlexrayCluster shall be referenced by a BusMirrorChannel in the System.]()

[constr_3473] BusMirrorChannelMappingFlexray.targetPduTriggering restriction [The FlexrayFrameTriggering of a Frame that contains a Pdu of which the PduTriggering is referenced by BusMirrorChannelMappingFlexray in the

role `targetPduTriggering` shall have the `allowDynamicLSduLength` attribute set to true.]()

6.10.3 Ethernet Destination Channel

[TPS_SYST_02205] **BusMirroring to Ethernet destination channel** [In case of CAN to Ethernet, LIN to Ethernet and FlexRay to Ethernet the `BusMirrorChannelMappingIp` meta-class shall be used for the modeling of the bus mirroring.]()

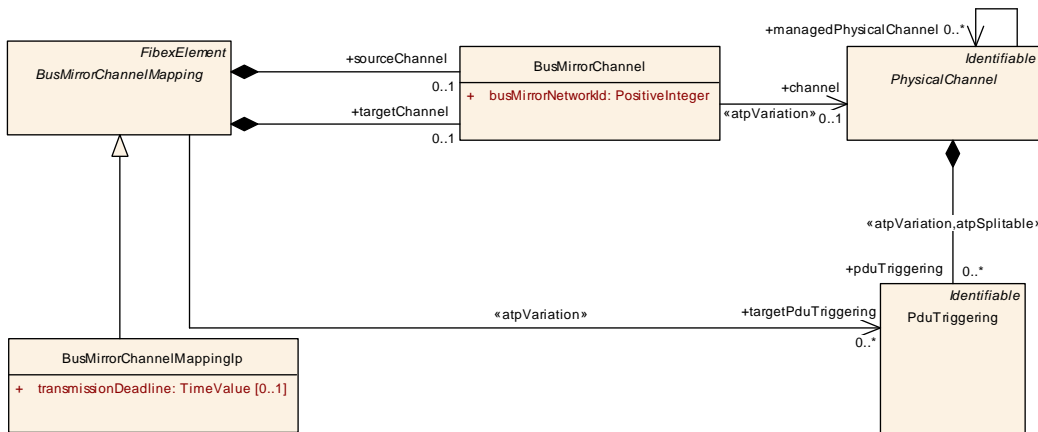


Figure 6.78: Bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel

Class	BusMirrorChannelMappingIp			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel. Tags: atp.recommendedPackage=BusMirrorChannelMappings			
Base	ARObject, BusMirrorChannelMapping , CollectableElement , FibexElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
transmissionDeadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table 6.274: BusMirrorChannelMappingIp

[constr_3474] **EthernetPhysicalChannel** as destination channel of **BusMirrorChannelMappingIp** [The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingIp` shall only reference an `EthernetPhysicalChannel` in the role `targetChannel`.]()

[constr_3475] **BusMirrorChannelMappingIp.targetPduTriggering** restriction [`BusMirrorChannelMappingIp` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.]()

6.10.4 User Defined Destination Channel

[TPS_SYST_02206] BusMirroring to UserDefined destination channel [In case of CAN to [UserDefinedPhysicalChannel](#), LIN to [UserDefinedPhysicalChannel](#) and FlexRay to [UserDefinedPhysicalChannel](#) the [BusMirrorChannelMappingUserDefined](#) meta-class shall be used for the modeling of the bus mirroring.]
()

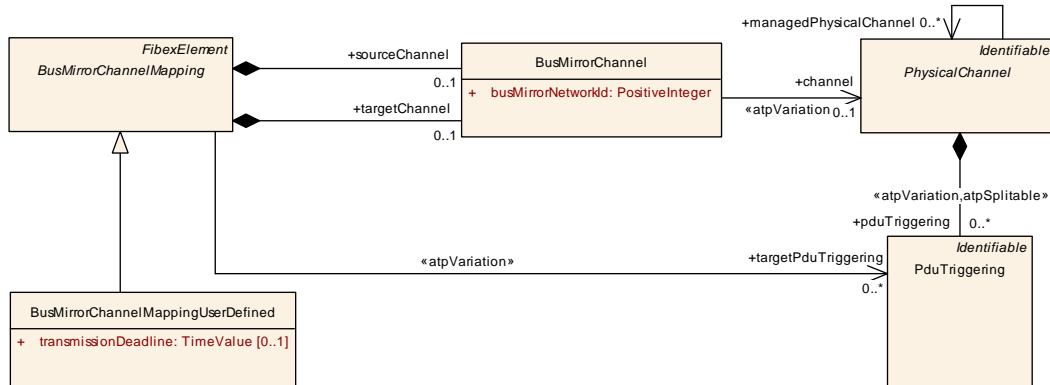


Figure 6.79: Bus mirroring between a CAN, LIN or FlexRay sourceChannel and a UserDefined targetChannel

Class	BusMirrorChannelMappingUserDefined			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a User Defined targetChannel. Tags: atp.recommendedPackage=BusMirrorChannelMappings			
Base	ARObject, BusMirrorChannelMapping , CollectableElement, FibexElement , Identifiable , Multilanguage Referrable, PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
transmission Deadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table 6.275: BusMirrorChannelMappingUserDefined

[constr_3476] UserDefinedPhysicalChannel as destination channel of BusMirrorChannelMappingUserDefined [The [BusMirrorChannel](#) that is aggregated by [BusMirrorChannelMappingUserDefined](#) shall only reference a [UserDefinedPhysicalChannel](#) in the role [targetChannel](#).] ()

[constr_3477] BusMirrorChannelMappingUserDefined.targetPduTriggering restriction [[BusMirrorChannelMappingUserDefined](#) is allowed to reference only one single [PduTriggering](#) in the role [targetPduTriggering](#).] ()

6.11 Fan-out

AUTOSAR supports three different fan-outs:

- Signal fan-out
- Pdu fan-out
- Frame fan-out

6.11.1 Signal fan-out

A Signal fan-out can either be RTE fan-out or COM Signal Gateway fan-out. The details are explained in the following subchapters.

6.11.1.1 RTE fan-out

The RTE supports a "signal fan-out" where one `SystemSignal` is sent in several `IPdus`.

[TPS_SYST_01109] RTE fan-out support [The RTE fan-out (signal fan-out) is described by the relation between `SystemSignal` and `ISignal`.

In the case of a "signal fan-out", several `ISignals` refer to the same `SystemSignal` as shown in the following example.]()

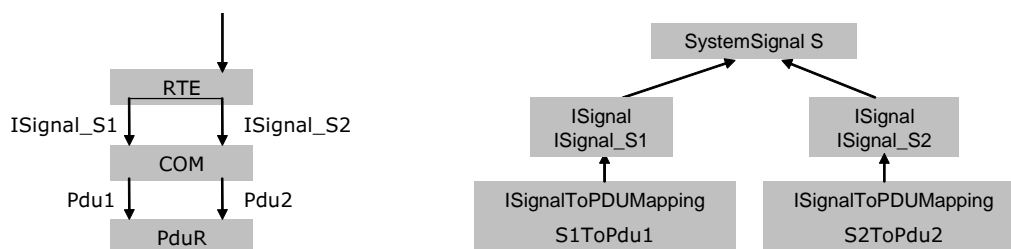


Figure 6.80: RTE fan-out

6.11.1.2 COM Signal Gateway fan-out

In Com [21] the Signal Gateway supports a fan-out where an incoming signal is routed to several destinations.

[TPS_SYST_01110] Com Signal Gateway fan-out support [A Signal Gateway fan-out (1:n routing) is described with the definition of several `ISignalMappings` in the `Gateway` description, which all refer to the same source `ISignalTriggering`.]()

Note that [constr_3514] applies for the relation between `ISignalToIPduMapping` to `ISignal`.

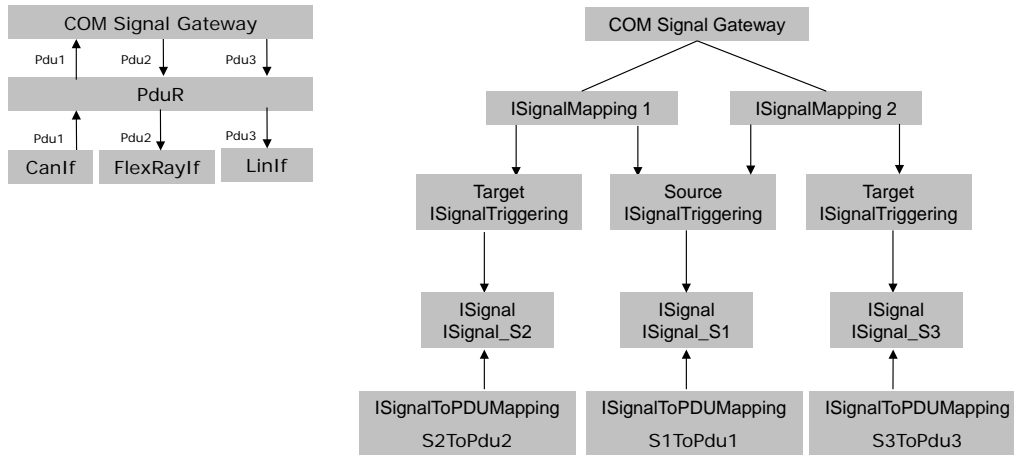


Figure 6.81: Com Gateway fan-out

6.11.2 Pdu fan-out

6.11.2.1 Pdu Router fan-out

The `Pdu Router` supports the "PDU fan-out" where one `IPdu` is sent to multiple destinations.

[TPS_SYST_01111] Pdu Router fan-out support [The `Pdu Router` fan-out is described by several `PduTriggering` elements pointing to the same `Pdu`³.

The sending ECU/PDU router has an output `IPduPort` that has the value of `communicationDirection` set to `out` and is referenced by the `PduTriggering`. According to the Cluster/Channel aggregation, the `Pdu Router` determines the clusters to use in its routing.]()

[TPS_SYST_01112] FlexrayCluster Pdu Router interaction [The following condition applies only in case of FlexRay on the same `FlexrayCluster` if two `PduTriggerings` refer to the same `Pdu`: this `Pdu` shall only be sent once to the FlexRay Interface. In other words the `Pdu Router` sends only one `Pdu Transmission` request to the FlexRay Interface.]()

³AUTOSAR Layered Architecture [15] defines which `Pdu` types are routed by the `Pdu Router`

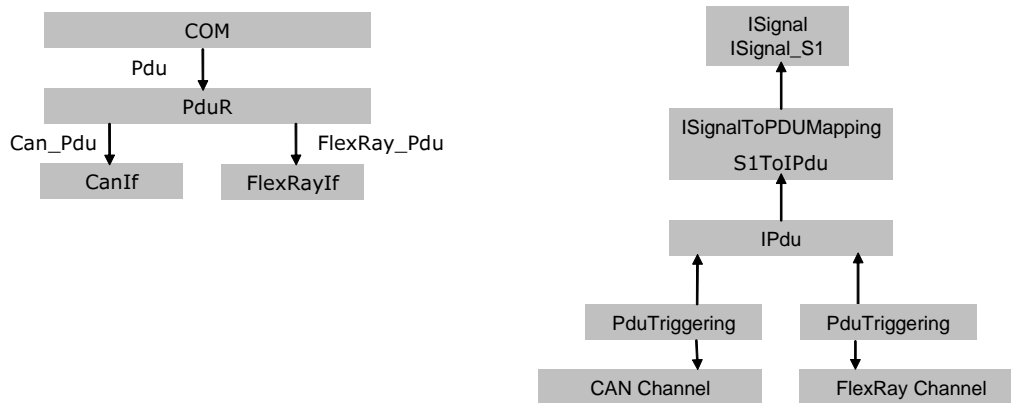


Figure 6.82: Pdu Router fan-out

6.11.2.2 Flexray Interface fan-out

The Flexray interface supports a fan-out where one `Pdu` is mapped into more than one frame on the same `CommunicationCluster`.

[TPS_SYST_01113] FlexRay Interface fan-out support [The redundant transmission in the FlexRay Interface in the static segment is described by

- one `FlexrayFrameTriggering` on each `PhysicalChannel`
- both `FlexrayFrameTriggerings` refer to the same `FlexrayFrame` with the same `Pdu`
- each `FlexrayFrameTriggering` aggregates the same number of `FlexrayAbsolutelyScheduledTimings`
- for every `FlexrayAbsolutelyScheduledTiming` on one `PhysicalChannel` a corresponding `FlexrayAbsolutelyScheduledTiming` with identical values shall be defined on the other `PhysicalChannel`

]()

If the fan-out is specified between different FlexRay channels of the same cluster it shall be handled by the FlexRay Interface.

The Flexray Interface does NOT handle fan-out/in between different clusters.

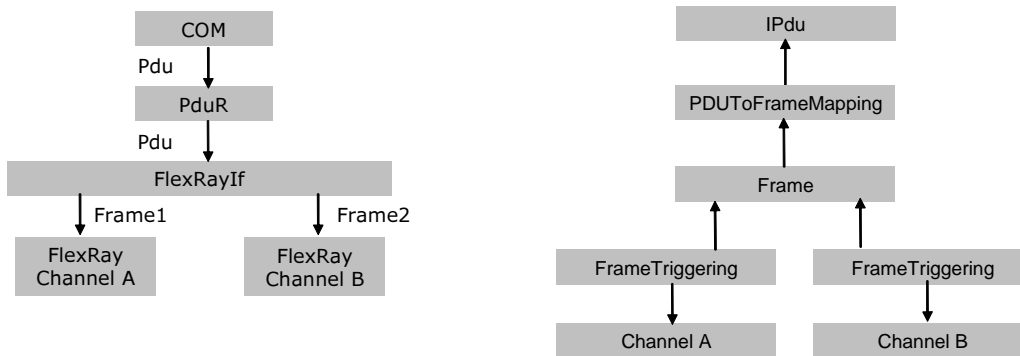


Figure 6.83: Bus Interface fan-out

6.11.3 Frame fan-out

[TPS_SYST_01114] **Frame fan-out support** [AUTOSAR supports the Frame fan-out only on the `FlexrayCluster` (see [TPS_SYST_01113]).]()

6.12 Log and Trace

`DltConfig` defines a Dlt module configuration for a specific `EcuInstance`. The name of the `EcuInstance` that is transported in the log and trace message can be configured by `DltConfig.dltEcuId`.

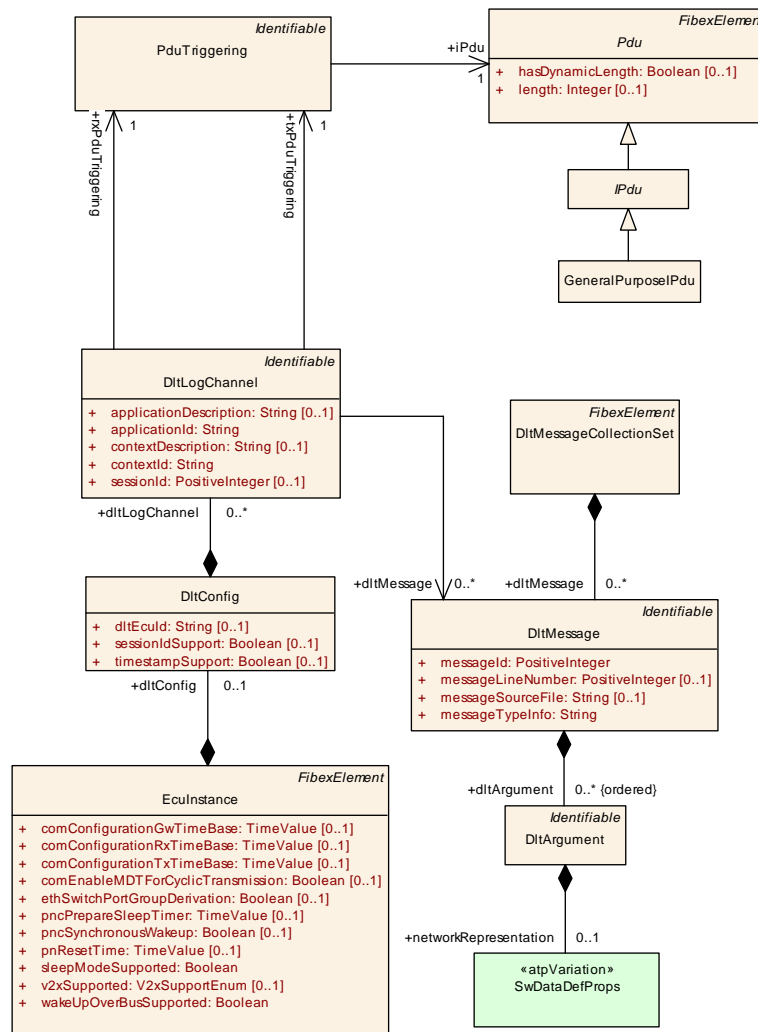


Figure 6.84: Dlt configuration of DltLogChannels and DltMessages

[TPS_SYST_02264] Usage of **DltLogChannel** [In each **DltConfig** the **PhysicalChannels** that are used to transport the log and trace messages are configured by the **DltLogChannel** elements.

Each **DltLogChannel** points to one **PduTriggering** in the role **txPduTriggering** to describe the Dlt Pdu that is transmitted by the **DltLogChannel**. The **rxPduTriggering** role is used to describe the Dlt Pdu that is received by the **DltLogChannel**.

The **DltLogChannel** defines the **applicationId** and **contextId** that are used to identify the source of the log and trace message.] ()

Class	DltConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Dlt			
Note	This element defines a Dlt configuration for a specific Ecu.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
dltEculd	String	0..1	attr	This attribute defines the name of the ECU for use within the Dlt protocol.
dltLogChannel	DltLogChannel	*	aggr	Describes the DltLogChannels that are configured for the log/trace message output
sessionIdSupport	Boolean	0..1	attr	This attribute defines whether the sessionId is used or not.
timestampSupport	Boolean	0..1	attr	This attribute defines whether a timestamp shall be added to the Dlt messages or not.

Table 6.276: DltConfig

Class	DltLogChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Dlt			
Note	This element contains the settings for the log/trace message output for a tuple of ApplicationId and ContextId (verbose mode) or a SessionId (non-verbose mode).			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
applicationDescription	String	0..1	attr	This attribute can be used to describe the applicationId that is used in the log and trace message in more detail.
applicationId	String	1	attr	This attribute identifies the SW-C/BSW module in the log and trace message.
contextDescription	String	0..1	attr	This attribute can be used to describe the contextId that is used in the log and trace message in more detail.
contextId	String	1	attr	This attribute is used to group log and trace messages produced by a SW-C/BSW modules to distinguish functionality (representing e.g. a library of the adaptive foundation linked into the application).
dltMessage	DltMessage	*	ref	Reference to DltMessages that can be transported over the DltLogChannel in the DltPdu.
rxPduTriggering	PduTriggering	1	ref	Reference to DltPdu that is received by the DltLog Channel
sessionId	PositiveInteger	0..1	attr	This attribute allows distinguishing log/trace messages from different instances of the same SW-C. It is required if sessionIdSupport of the aggregating DltConfig is True.
txPduTriggering	PduTriggering	1	ref	Reference to DltPdu that is transmitted by the DltLog Channel.

Table 6.277: DltLogChannel

[constr_5097] [DltLogChannel.txPduTriggering](#) and [DltLogChannel.rxPduTriggering](#) shall point to **GeneralPurposeIPdus** of category **DLT** [[DltLogChannel](#) shall only reference [PduTriggerings](#) that are pointing to [GeneralPurposeIPdus](#) of category **DLT** in the roles [txPduTriggering](#) and [rxPduTriggering](#).]()

A Log message may be sent in two modes: either **VERBOSE** or **NON-VERBOSE** mode.

In the VERBOSE mode the payload of the log message contains values together with type information such that the external client can interpret the received data.

In NON-VERBOSE mode the payload of the log message contains a unique message ID and some data. The description of the payload layout according to the corresponding `messageId` is described by the `DltMessage` element.

Class	DltMessageCollectionSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Dlt			
Note	Collection of DltMessages Tags: atp.recommendedPackage=DltMessageCollectionSets			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
dltMessage	DltMessage	*	aggr	Collection of DltMessages in the DltMessageCollection Set.

Table 6.278: DltMessageCollectionSet

Class	DltMessage			
Package	M2::AUTOSARTemplates::SystemTemplate::Dlt			
Note	This element defines a DltMessage.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
dltArgument (ordered)	DltArgument	*	aggr	Ordered collection of DltArguments in the DltMessage.
messageId	PositiveInteger	1	attr	This attribute defines the unique Id for the DltMessage.
messageLine Number	PositiveInteger	0..1	attr	This attribute describes the position in the source file in which this log message was called.
messageSource File	String	0..1	attr	This attribute describes the source file in which this log message was called.
messageType Info	String	1	attr	This attribute describes the message Type

Table 6.279: DltMessage

Class	DltArgument			
Package	M2::AUTOSARTemplates::SystemTemplate::Dlt			
Note	This element defines an Argument in a DltMessage.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
network Representation	SwDataDefProps	0..1	aggr	Definition of the networkRepresentation of the Dlt Argument.

Table 6.280: DltArgument

[constr_5098] Usage of `DltArgument.networkRepresentation` [Usage of `DltArgument.networkRepresentation` shall follow the restrictions given in table 6.281.]()

Attributes of SwDataDefProps	networkRepresentationProps
additionalNativeTypeQualifier	NA
annotation	NA
baseType	D
compuMethod	NA
dataConstr	NA
displayFormat	NA
displayPresentation	NA
implementationDataType	NA
invalidValue	NA
swAddrMethod	NA
swAlignment	NA
swBitRepresentation	NA
swCalibrationAccess	NA
swCalprmAxisSet	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. swCalprmRef	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. swVariableRef	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. sharedAxisType	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. inputVariableType	NA
swCalprmAxisSet/ AxisIndividual/ Unit	NA
swCalprmAxisSet/ BaseType	NA
swComparisonVariable	NA
swDataDependency	NA
swHostVariable	NA
swImpIPolicy	NA
swIntendedResolution	NA
swInterpolationMethod	NA
swIsVirtual	NA
swPointerTargetProps	NA
swRecordLayout	NA
swRefreshTiming	NA
swTextProps	NA
swValueBlockSize	NA
unit	D
valueAxisDataType	NA

Table 6.281: Allowed SwDataDefProps Attributes for the networkRepresentation of DltArguments

The following settings apply in table 6.281:

D Attribute can be **defined** in the scope of this element.

NA Attribute is **not applicable** for usage in the scope of this element.

[constr_5099] Standardized values of `DltMessage.messageTypeInfo` [The following values of attributes `DltMessage.messageTypeInfo` are standardized by AUTOSAR:

- DLT_LOG_OFF
- DLT_LOG_FATAL
- DLT_LOG_ERROR
- DLT_LOG_WARN
- DLT_LOG_INFO
- DLT_LOG_DEBUG
- DLT_LOG_VERBOSE
- DLT_TRACE_VARIABLE
- DLT_TRACE_FUNCTION_IN
- DLT_TRACE_FUNCTION_OUT
- DLT_TRACE_STATE
- DLT_TRACE_VFB
- DLT_NW_TRACE_IPC
- DLT_NW_TRACE_CAN
- DLT_NW_TRACE_FLEXRAY
- DLT_NW_TRACE_MOST
- DLT_NW_TRACE_ETHERNET
- DLT_NW_TRACE_SOMEIP
- DLT_NW_TRACE_0x7
- DLT_NW_TRACE_0x8
- DLT_NW_TRACE_0x9
- DLT_NW_TRACE_0x10
- DLT_NW_TRACE_0x11
- DLT_NW_TRACE_0x12
- DLT_NW_TRACE_0x13
- DLT_NW_TRACE_0x14
- DLT_NW_TRACE_0x15
- DLT_CONTROL_REQUEST

- DLT_CONTROL_RESPONSE

]0

6.13 Support of Complex Drivers

The System Template allows the integration of custom communication means into AUTOSAR [EcuInstances](#).

[TPS_SYST_01115] CDD communication support [The elements [UserDefinedPdu](#) and [UserDefinedIPdu](#) shall be used to describe the Pdu-based communication via Complex Drivers.] ([RS_SYST_00043](#))

The [UserDefinedPdu](#) and [UserDefinedIPdu](#) elements are described in chapter [6.3](#) in more detail.

The [UserDefinedIPdu](#) can be used to describe the communication if a new BSW module was added above the PduR, e.g a Diagnostic Service.

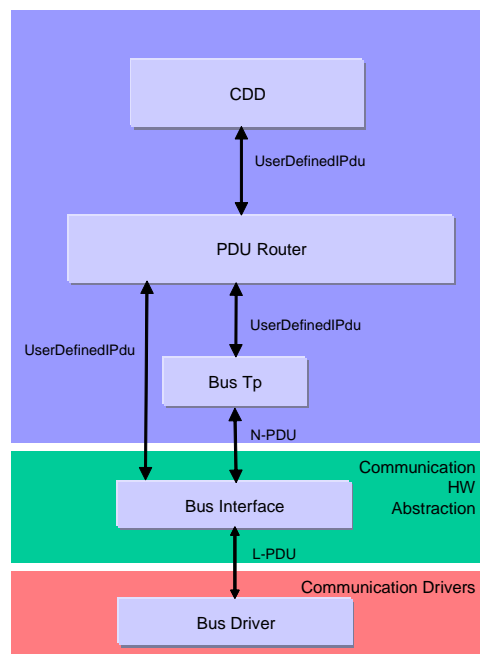


Figure 6.85: CDD over PduR

The [UserDefinedPdu](#) can be used to describe the communication if a new BSW module was added above an Interface, e.g. a new Nm module or XCP.

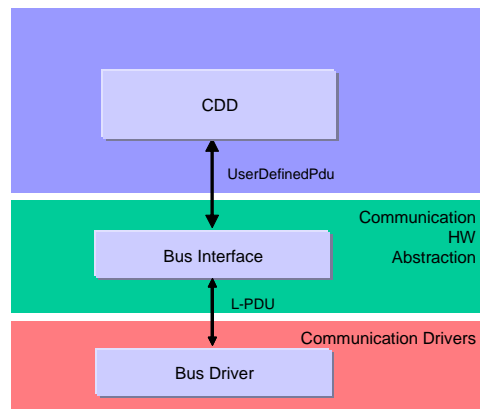


Figure 6.86: CDD over Bus Interface

6.14 MetaData in SenderReceiverInterface

As described in [5], there is the ability for the application software to unlock access to `Pdu` meta-data by using the aggregation `SenderReceiverInterface.metaDataItemSet`.

This modeling allows for the arbitrary definition of meta-data semantics that - of course - has to be supported by the `Pdu` where a specific `dataElement` is finally mapped to.

This consistency is ensured by deriving the `Pdu` meta-data from the modeling in `SenderReceiverInterfaces`. However, this approach can only work if all `dataElements` mapped to the given `Pdu` define the same meta-data structure or do not define the usage of meta-data at all.

[constr_5100] Compatibility of two `MetaDataItemSets` [Under the condition that sender and receiver typed by a `SenderReceiverInterface` use meta-data and are mapped to the same `EcuInstance` the following condition applies: two `MetaDataItemSets` are compatible if all of the following conditions are fulfilled:

- They aggregate the same number of `MetaDataItems`.
- The value of `MetaDataItem.length` of corresponding `MetaDataItems` is identical.
- The value of `MetaDataItem.metaDataItemType` of corresponding `MetaDataItems` is identical.

]()

[constr_5101] Consistent Definition of meta-data [If the `dataElement` referenced by a `SenderReceiverToSignalMapping` is also referenced by a `MetaDataItemSet` in the role `dataElement` and the mapping via `SystemSignal`, `ISignal`, and `ISignalToIPduMapping` down to an `ISignalIPdu` exists then all other `dataElements` that are also mapped to the same `ISignalIPdu` shall either

- not be referenced by a `MetaDataItemSet` in the role `dataElement` (i.e. does not make use of meta-data) or
- the definition of meta-data in the context of the affected `SenderReceiverInterfaces` is compatible (according to the definition of compatible specification of meta-data described in `[constr_5100]`).

}]()

6.15 Signal Service Translation

AUTOSAR Adaptive Platform restricts communication paradigm to Service-oriented communication. A major part of the vehicle however still uses Signal-based communication means, therefore a translation of these two approaches has to be performed.

One major goal of AUTOSAR is that it covers both

- high-performance microprocessor Machines (connected via high payload and high bandwidth Ethernet networks)
- highly embedded microcontroller ECUs (connected via ethernet, but also via low payload and low bandwidth CAN and LIN networks).

A seamless development of a vehicle shall be supported with one AUTOSAR methodology. Therefore a translation is required which closes the gap between

- Signal-based communication on Classic platform
- Service-oriented communication on Adaptive platform.

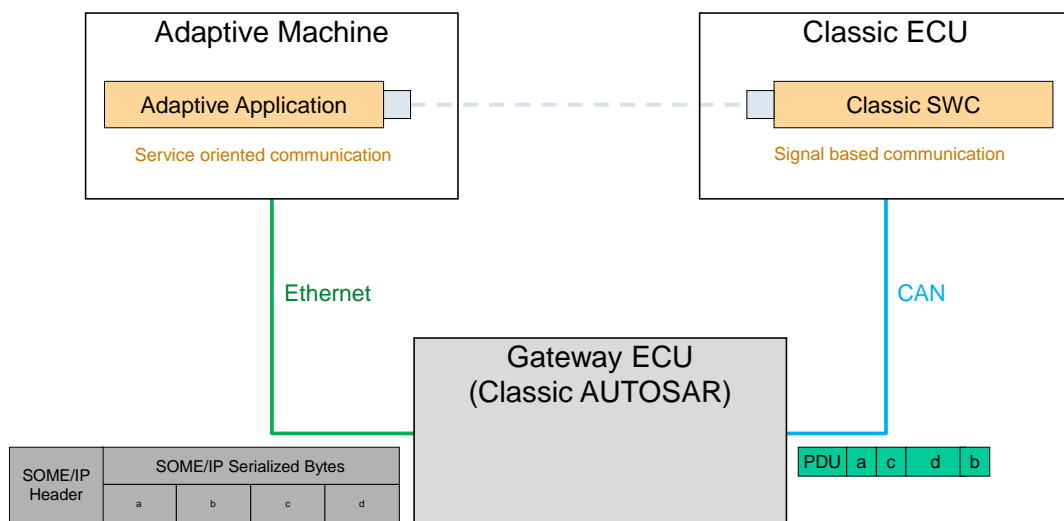


Figure 6.87: Signal Service Translation in Classic platform gateway ECU

The goal of this chapter is to standardize a translation between the two communication transport configurations on an AUTOSAR Classic platform ECU. A similar approach is

also provided in the Manifest specification for the Adaptive platform [27]. It is up to the vehicle architecture design to choose whether the signal service translation shall be implemented on a Classic platform ECU or on an Adaptive platform Machine.

6.15.1 Architectural setup

The implementation of the signal service translation on the Classic platform is done in an Application Software Component above the RTE. This applies for *events* and *field notifiers*. Methods are handled separately (see section 6.15.1.1).

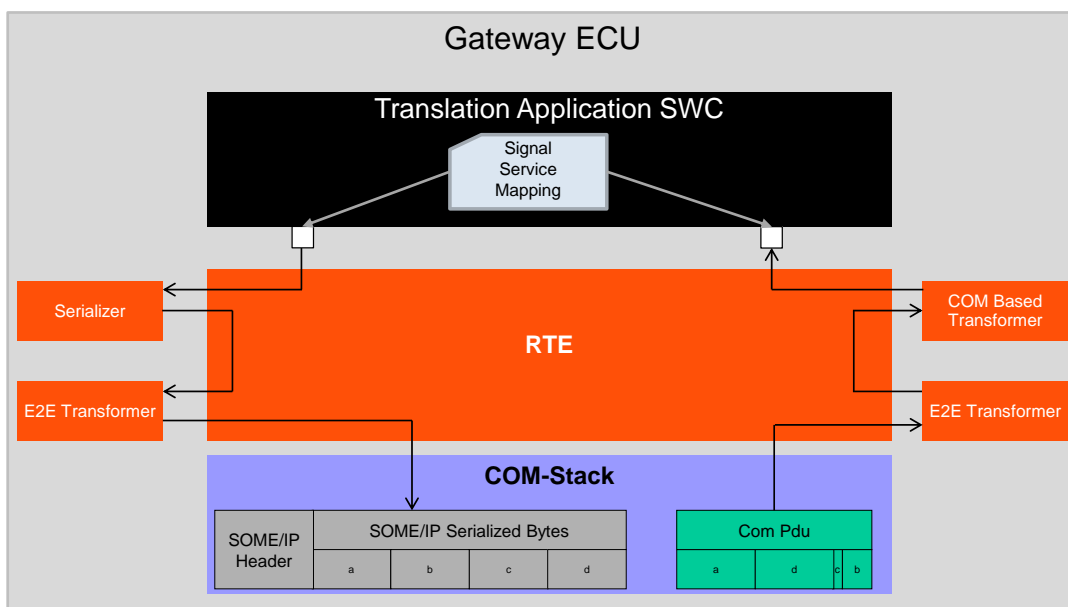


Figure 6.88: Signal Service Translation Application Software Component

For the signal-based part the full functionality of the Classic platform COM-Stack is available and may be configured such that the signal-based *ISignalIPdus* may originate from a variety of sources (Can, Lin, FlexRay) and the *ISignalIPdus* may be safety and security protected.

For the service-oriented part it has to be guaranteed that the defined SOME/IP Service actually is compatible to the Adaptive platform. This applies for the payload part (e.g. the SOME/IP serializer has to be used) as well as for the control path using *BswM* and *ServiceDiscovery*.

The behavioral part of the Translation Software Component itself defines how the data from signal-based side is transported to the service-oriented side, and vice versa (see section 6.15.4).

6.15.1.1 Method handling

If a translated service has `methods` (or `getter/setter` calls of `fields`) defined, the payload is already serialized according to the SOME/IP serialization rules. So even on a `CanFD` and `FlexRay` network the transport is possible.

The assumption is that on the *signal-based* side even the `method` messages have a SOME/IP Transformer [28] header consisting of the 32-bit *Request ID* and the 32-bit *Protocol Version, Interface Version, Message Type, and Return Code* as illustrated in figure 6.89. As the SOME/IP Header parts *Message ID* and *Length* are added by the *Socket Adapter* these parts are not available on the *signal-based* side.

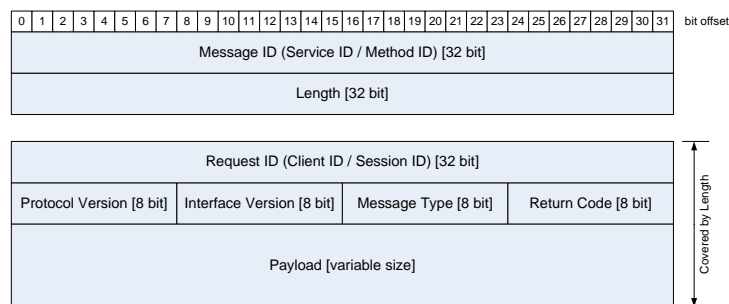


Figure 6.89: SOME/IP Header structure

By defining an `IPduMapping` (PDU based routing functionality as defined in section 8.2) the method message (call and reply) can be routed between the *signal-based* and the SOME/IP based networks (shown in figure 6.90). The configuration of these `IPduMappings` is out of scope of the signal-service-translation.

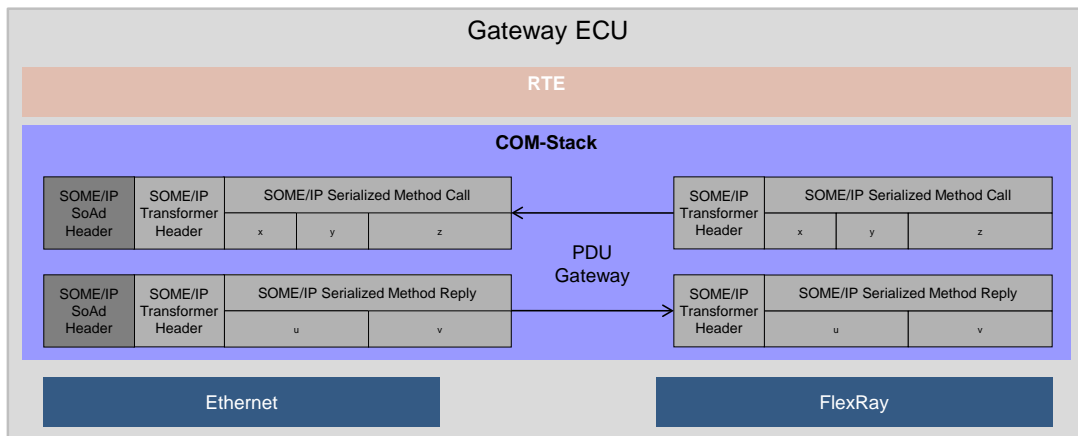


Figure 6.90: Direct routing of Method messages

For the runtime aspects: As the routing paths are dynamically controlled using the service discovery control as defined in section 6.15.3 the *PduR* will only route those messages which are activated. Thus, if the respective translated service is not active, the directly routed method messages are not routed.

6.15.2 Mapping description

Signal Service Translation is used to alter the serialization representation of data to be compatible with the respective transport network. I.e. on an Ethernet network a SOME/IP serialized data representation is mostly suitable, while on a CAN network the packed signal-based data representation often required due to the low payload data size available.

As indicated in section 6.15.1 the implementation of the translation shall be done in an Application Software Component above the RTE. For the definition of the intended mapping and behavior however the `CompositionSwComponentType` is used. This allows to represent the requirements on the translation and still allow some freedom with respect to the actual implementation later on.

The element which defines the behavioral aspects of the signal-service-translation is the meta-class `SignalServiceTranslationProps`. The references to the `VariableDataPrototype` in the role `translationTarget` define to which events the `SignalServiceTranslationProps` apply. For this reference a `VariableDataPrototypeInSystemInstanceRef` is used (see also chapter B.8).

- In case of signal-to-service translation the `SignalServiceTranslationProps.translationTarget` collect all resulting events which belong to one provided service instance.
- In case of service-to-signal translation the `SignalServiceTranslationProps.translationTarget` collect all resulting signals which are translated from one service instance.

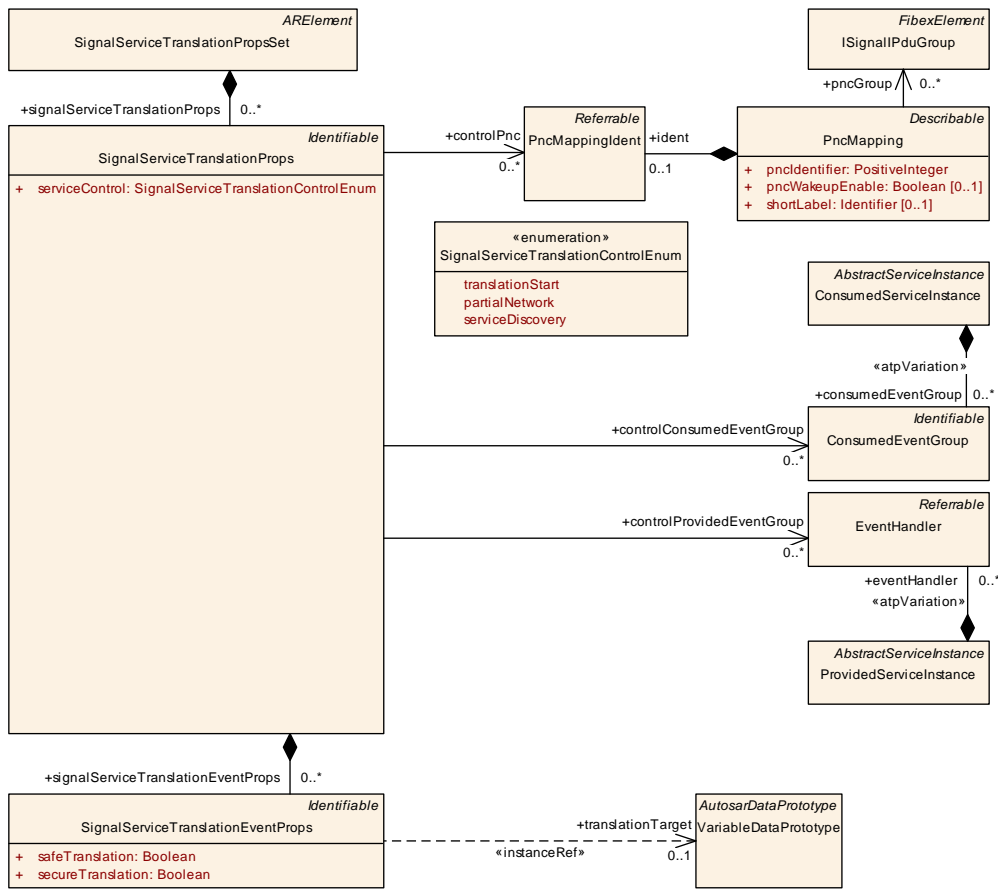


Figure 6.91: Signal Service Translation properties

Class	SignalServiceTranslationPropsSet			
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation			
Note	Collection of SignalServiceTranslationProps. Tags: atp.Status=draft atp.recommendedPackage=SignalServiceTranslationProps			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
signalServiceTranslationProps	SignalServiceTranslationProps	*	aggr	Collection of SignalServiceTranslationProps.

Table 6.282: SignalServiceTranslationPropsSet

Class	SignalServiceTranslationProps			
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation			
Note	This element allows to define the properties which are applicable for the signal-service-translation service. Tags: atp.Status=draft			





Class		SignalServiceTranslationProps		
Base		ARObject, Identifiable , MultilanguageReferrable , Referrable		
Attribute	Type	Mult.	Kind	Note
controlConsumedEventGroup	ConsumedEventGroup	*	ref	Reference to the EventGroup which encapsulates the signal-based payload. Tags: atp.Status=draft
controlPnc	PncMappingIdent	*	ref	Reference to the PNCs which control the offer/subscribe behavior of the translated service instance. Tags: atp.Status=draft
controlProvidedEventGroup	EventHandler	*	ref	Reference to the provided event groups (aka Event Handler) which control the availability of the service instance. Tags: atp.Status=draft
serviceControl	SignalServiceTranslationControlEnum	1	attr	Defines how the service instance control shall behave.
signalServiceTranslationEventProps	SignalServiceTranslationEventProps	*	aggr	Defines properties for a single translated event. Tags: atp.Status=draft

Table 6.283: SignalServiceTranslationProps

Class		SignalServiceTranslationEventProps		
Package		M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation		
Note		This element allows to define the properties which are applicable for the signal-service-translation event. Tags: atp.Status=draft		
Base		ARObject, Identifiable , MultilanguageReferrable , Referrable		
Attribute	Type	Mult.	Kind	Note
elementProps	SignalServiceTranslationElementProps	*	aggr	Defines properties for a single translated element. Tags: atp.Status=draft
safeTranslation	Boolean	1	attr	Defined whether the translation shall happen in a safe way.
secureTranslation	Boolean	1	attr	Defined whether the translation shall happen in a secure way.
translationTarget	VariableDataPrototype	0..1	iref	Reference to a VariableDataPrototype representing the target of signal-service-translation.

Table 6.284: SignalServiceTranslationEventProps

A simple Signal Service Translation setup is to have a one-to-one correspondence between the signal-based data definition and the service-oriented data definition. This is illustrated in figure 6.92.

Here the setup allows for a simple [PassThroughSwConnector](#) because the involved [PortInterfaces](#) are identical.

Please consult with [constr_1248] in the Software Component Template [5] for details on the compatibility of connected [PortInterfaces](#).

Note that the translated *P1.dataX* and *P2.dataY* may be configured in the COM-Stack to be events that belong to one service instance or to belong to two different service

instances. The setup in the example of figure 6.92 uses two `SignalServiceTranslationProps` elements, indicating that `P1.dataX` belongs a different service instance than `P2.dataY`.

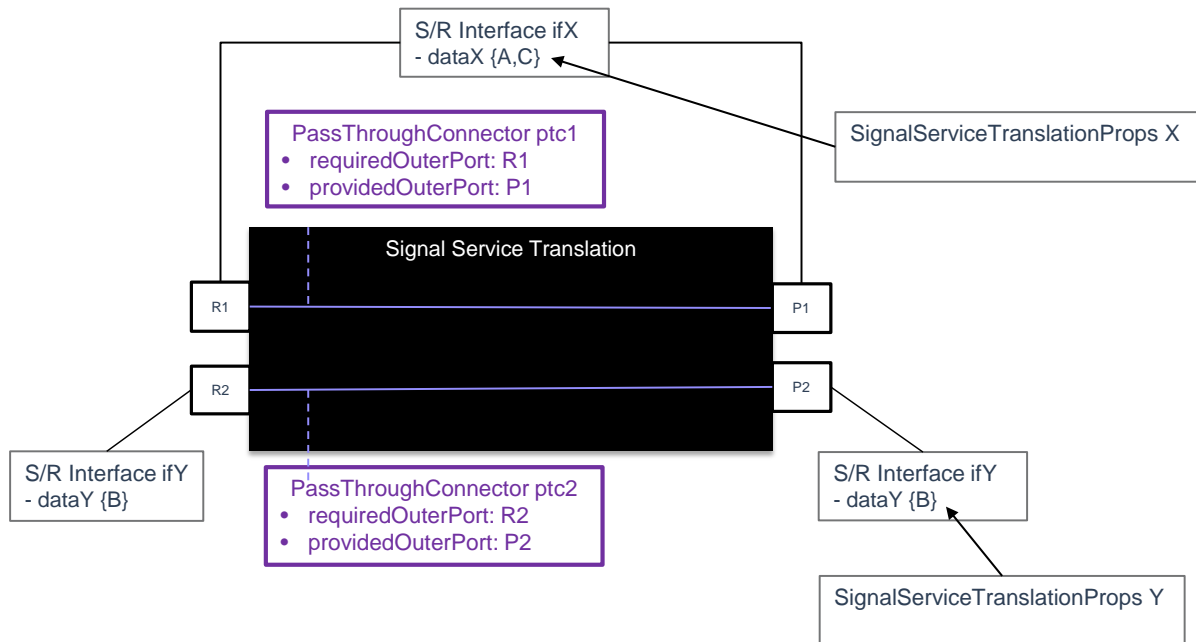


Figure 6.92: Mapping description using only PassThroughSwConnectors

The usage of the `PassThroughSwConnector` and (optionally) the accompanying `PortInterfaceMapping` already suffice to describe the intended mapping from a structural point of view:

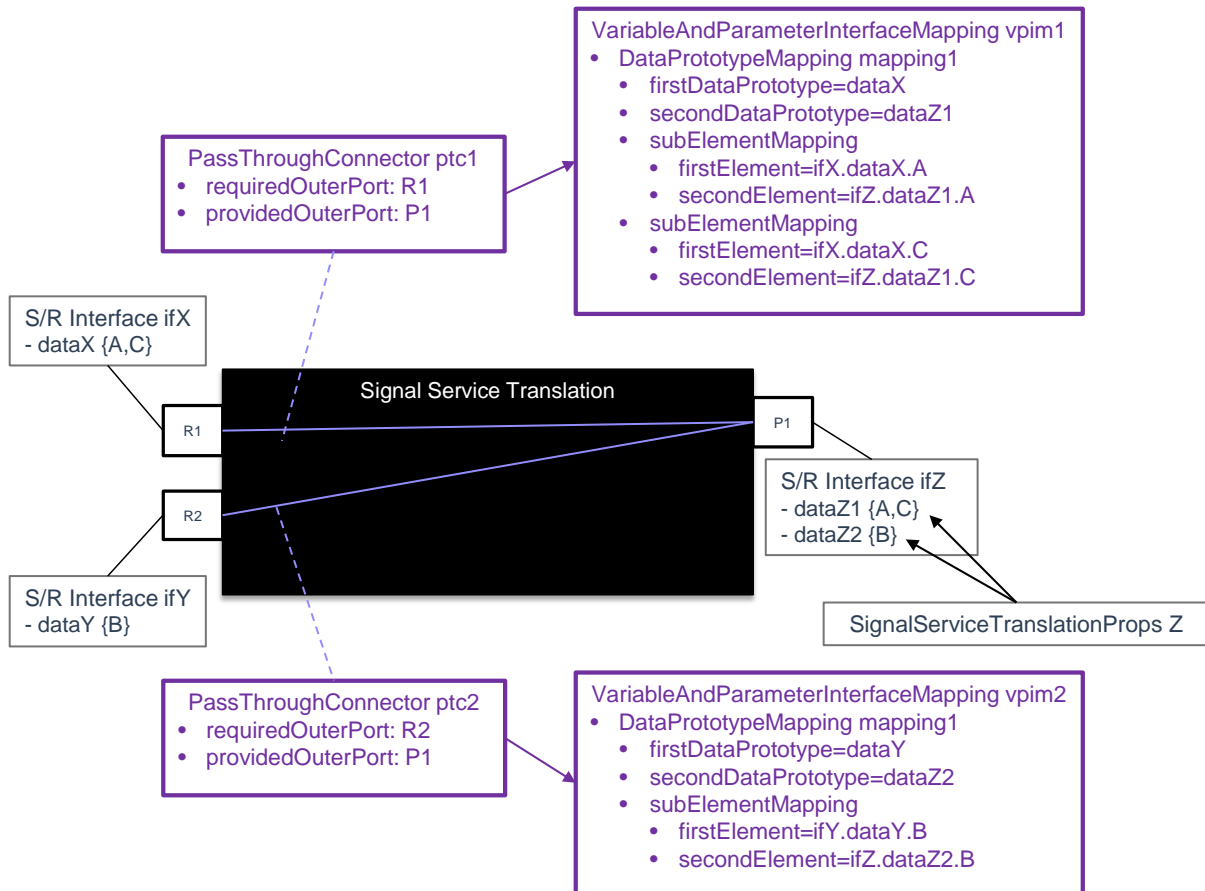


Figure 6.93: Mapping description using PassThroughSwConnectors and PortInterfaceMappings

In figure 6.93 the additional usage of `PortInterfaceMappings` is required because the connected `PortInterfaces` are not identical (in this example have different `shortNames`).

In this example there is only one `SignalServiceTranslationProps` element defined which points to both the data elements `dataZ1` and `dataZ2`. Thus the translated events belong to the same service instance.

Although the figure 6.93 indicates that the output Port `P1` data `dataZ1` and `dataZ2` are composed of input from several sources, this is not true for the individual data elements. `dataZ1` is solely composed out of `dataX`, while `R1.dataZ2` is composed out of `R2.dataY`. Thus the example shown is still a mapping from *one source*.

6.15.2.1 Mapping description from several sources

The mapping setup in figure 6.94 shows an example of a true multi-source mapping definition: The content of `dataZ3` is composed out of several sources: `dataX` and `dataY`.

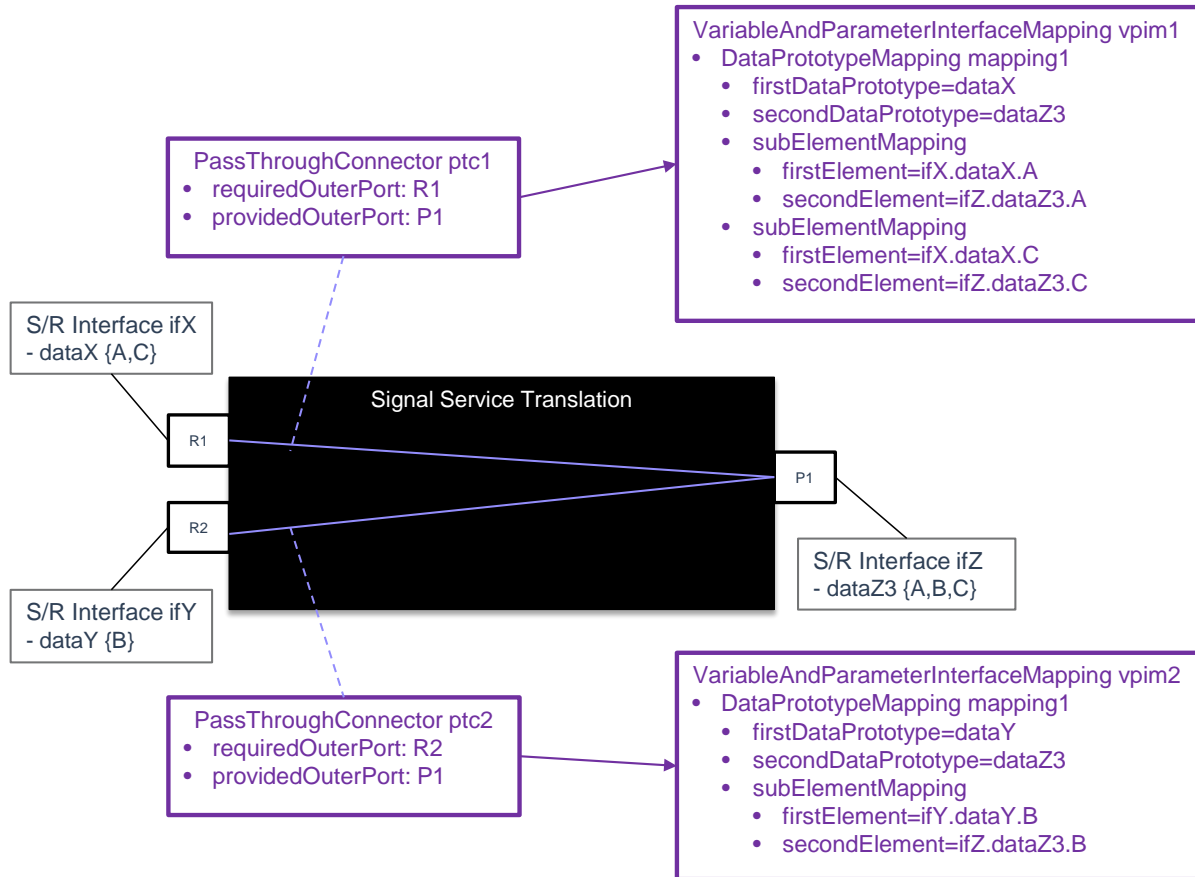


Figure 6.94: Mapping description using PassThroughSwConnectors and PortInterfaceMappings from several sources

[TPS_SYST_03038]{DRAFT} Definition of transmission triggers for translations with different sources [The attribute `SignalServiceTranslationElementProps.transmissionTrigger` defines which translation parts contribute to the transmission triggering for the mapped payload.]([RS_SYST_00059](#))

[TPS_SYST_03039]{DRAFT} Full translation before transmission triggering [In case there has been a transmission trigger caused by a source signal the signal-service-translation shall process all other mapped source signals from the triggering source context (signal group or IPdu) before actually sending out the target.]([RS_SYST_00059](#))

[TPS_SYST_03040]{DRAFT} Transmission trigger for translations with different sources [If the attribute `SignalServiceTranslationElementProps.transmissionTrigger` equals `true` then the reception of the respective source signal does cause the sending of the target (after all mapped sources from the same source context have been translated, see [TPS_SYST_03039]).]([RS_SYST_00059](#))

[TPS_SYST_03041]{DRAFT} No transmission trigger for translations with different sources [If the attribute `SignalServiceTranslationElementProps.transmissionTrigger` is not defined or has the value `false` then

the reception of the respective source signal does not cause the sending of the target.]
(RS_SYST_00059)

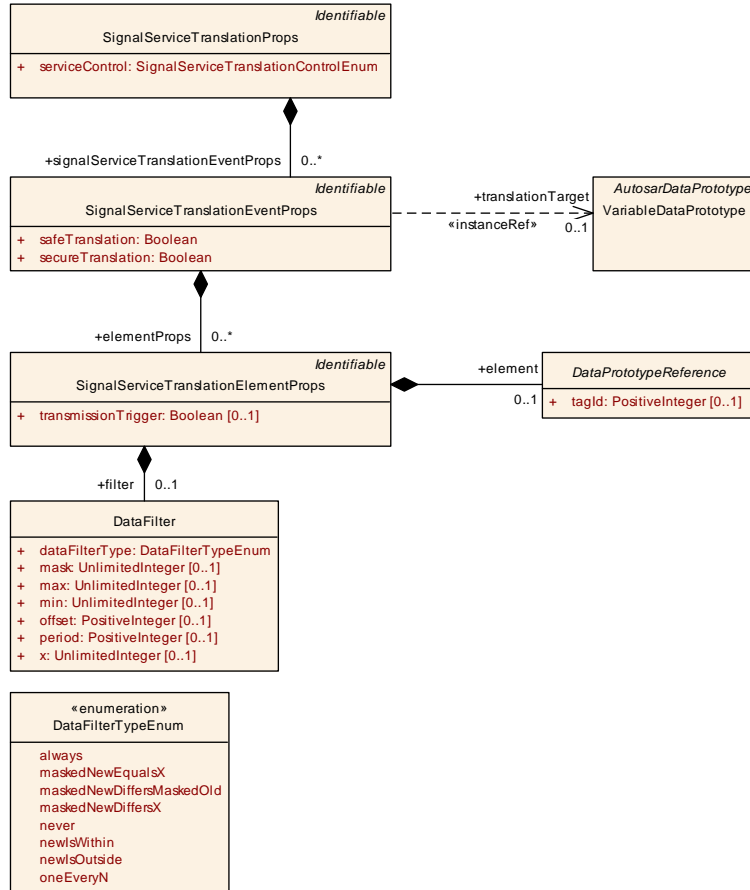


Figure 6.95: Signal Service Translation element properties

Class	SignalServiceTranslationElementProps			
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation			
Note	Defined translation properties for individual mapped elements. Tags: atp.Status=draft			
Base	ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
element	DataPrototypeReference	0..1	aggr	Reference to the leaf element the SignalServiceTranslationElementProps apply to. Tags: atp.Status=draft
filter	DataFilter	0..1	aggr	Defines an optional filter to be applied during translation. Tags: atp.Status=draft
transmission Trigger	Boolean	0..1	attr	Defines whether the source element (which is mapped to the referenced element) triggers the sending of the respective payload.

Table 6.285: SignalServiceTranslationElementProps

6.15.2.2 Mapping description and data conversion

As the mapping for signal-service-translation is implemented on Application level also the data conversion possibilities of the RTE may be used directly. [TPS_SWCT_01560] and [TPS_SWCT_01561] in the Software Component Template [5] define for which *category*s of *CompuMethods* data conversion is supported.

6.15.3 Service discovery control

The service discovery module [29] handles the offering/finding of service instances as well as the subscription to event groups. The behavior for each service instance may be controlled by Application software components via the BswM.

In scope of the Signal Service Translation the Translation Software Component needs to take control of the offering and subscribing to service instances. The general setup is illustrated in figure 6.96.

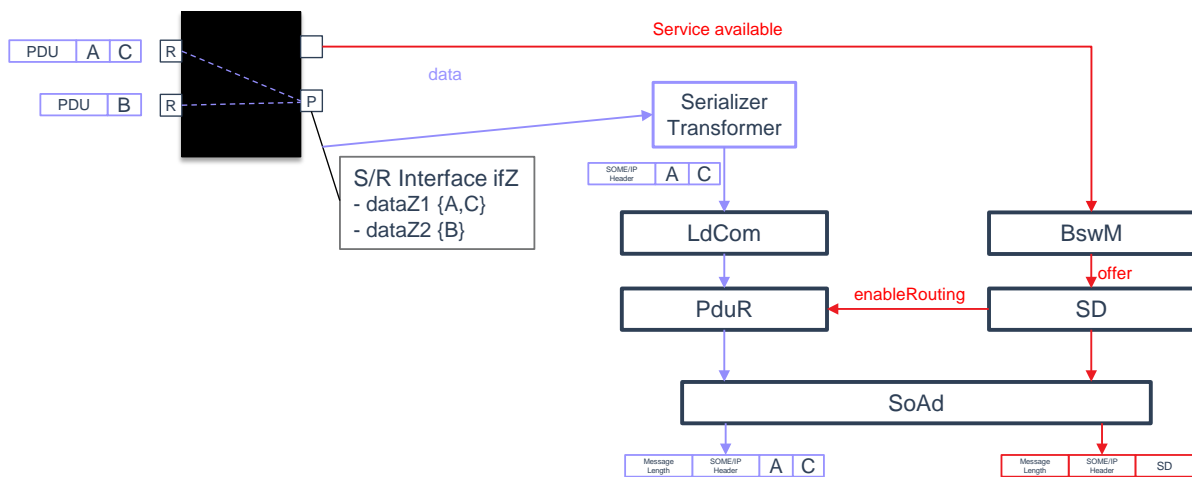


Figure 6.96: Interaction between translation SWC and Service Discovery

At which point in time a specific service instance (originating from signal service translation) is actually offered / subscribed at the service discovery can be defined per service instance:

Possible approaches for service availability/subscription are:

- *translationStart* - start of ECU (see section 6.15.3.1)
- *partialNetwork* - availability of involved partial networks (see section 6.15.3.2)
- *serviceDiscovery* - availability of related service instance (see section 6.15.3.3)

The attribute *SignalServiceTranslationProps.serviceControl* defines the service instance control behavior.

Enumeration	SignalServiceTranslationControlEnum
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation
Note	This enumeration allows to define how the service instance offer/subscribe control shall behave. Tags: atp.Status=draft
Literal	Description
partialNetwork	Defines the start of service control when specific partial networks are active. Tags: atp.EnumerationLiteralIndex=1
serviceDiscovery	Defines the start of service control when other service is available. Tags: atp.EnumerationLiteralIndex=2
translationStart	Defines the start of service control at translation start. Tags: atp.EnumerationLiteralIndex=0

Table 6.286: SignalServiceTranslationControlEnum

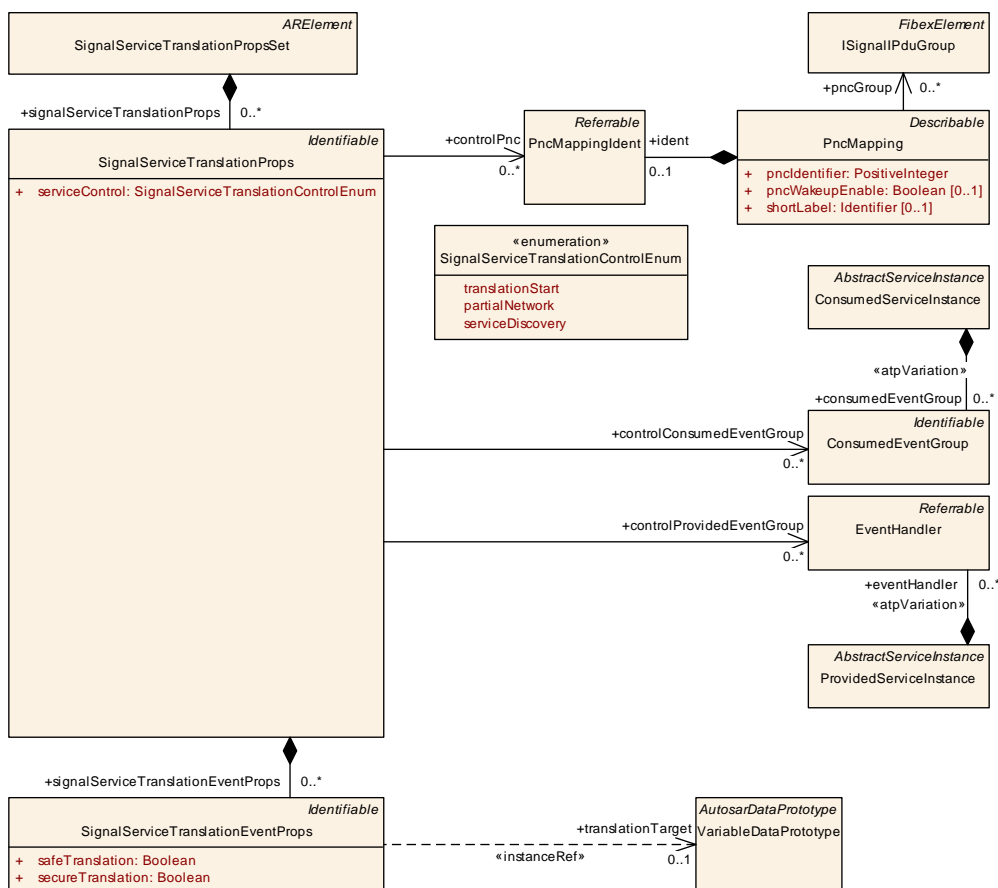


Figure 6.97: Signal Service Translation control

6.15.3.1 Service control at ECU start

The approach of service control at ECU start is to utilize the *AutoAvailable / AutoRequire* feature of the service discovery module where the service instance is automatically offered at startup of the ECU. In this case the translation software component

does not need to interact with the BswM to control the service state of each individual service instance.

It is configurable per translated service event whether its shall be automatically controlled at ECU start.

[TPS_SYST_03022]{DRAFT} Provided service instance with `translationStart` requires `SdServerServiceAutoAvailable` [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart` then the respective translated service instance shall be configured to use the `SdServerServiceAutoAvailable` feature.]([RS_SYST_00059](#))

[TPS_SYST_03023]{DRAFT} Required service instance with `translationStart` requires `SdClientServiceAutoRequire` [For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart` then the respective translated service instance shall be configured to use the `SdClientServiceAutoRequire` feature.]([RS_SYST_00059](#))

[TPS_SYST_03024]{DRAFT} Required service instance with `translationStart` requires `SdConsumedEventGroupAutoRequire` [For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart` then all required translated event groups shall be configured to use the `SdConsumedEventGroupAutoRequire` feature.]([RS_SYST_00059](#))

6.15.3.2 Service control due to availability of partial networks

If the availability of the signal-based PDUs is controlled using Partial Networking then the respective translated services offers/subscriptions can only be activated if the specific partial networks are active. This relationship is used to control the availability of specific service instances.

[constr_3545]{DRAFT} Mandatory reference to a `Pnc` in case of `partialNetwork` [If the `SignalServiceTranslationProps.serviceControl` equals `partialNetwork` then the reference `SignalServiceTranslationProps.controlPnc` shall point to at least one `PncMappingIdent`.]()

[TPS_SYST_03025]{DRAFT} Control of service instance in case of `partialNetwork` [If the `SignalServiceTranslationProps.serviceControl` equals `partialNetwork` then the respective service instance shall be configured to be controlled using the BswM/SD configuration.]([RS_SYST_00059](#))

[TPS_SYST_03026]{DRAFT} Monitoring of the `partial networks` status in case of `partialNetwork` for provided service instance [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `partialNetwork` then the translation software component shall monitor the status of the referenced `partial networks` (referenced via `SignalServiceTranslationProps.controlPnc`).

If at least one of the referenced *partial networks* is in the state *requested* then the translation software component shall *offer* the respective translated service instance.](RS_SYST_00059)

[TPS_SYST_03027]{DRAFT} Monitoring of the *partial networks* status in case of *partialNetwork* for required service instance [For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `partialNetwork` then the translation software component shall monitor the status of the referenced *partial networks* (referenced via `SignalServiceTranslationProps.controlPnc`).

If at least one of the referenced *partial networks* is in the state *requested* then the translation software component shall *find* the respective translated service instance and *subscribe* to its event groups.](RS_SYST_00059)

6.15.3.3 Service control due to availability of related service instance

There are scenarios where the signal-based PDUs are actually controlled using SOME/IP Service Discovery. So there are services defined using events/methods/fields and the service instances are offered / subscribed using the SOME/IP Service Discovery, just the payload of such services is not serialized according to the SOME/IP transformer rules. Therefore a signal service translation is required for the payload.

[constr_3546]{DRAFT} Mandatory reference to a *ConsumedEventGroup* in case of *serviceControl* [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` then the reference `SignalServiceTranslationProps.controlConsumedEventGroup` shall point to at least one `ConsumedEventGroup`.]()

[constr_3547]{DRAFT} Mandatory reference to an *EventHandler* in case of *serviceControl* [For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` then the reference `SignalServiceTranslationProps.controlProvidedEventGroup` shall point to at least one `EventHandler`.]()

[TPS_SYST_03028]{DRAFT} Initial *service find* for a provided service instance with *serviceControl* [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` then the translation software component shall upon startup issue a *service find* for the `ConsumedServiceInstance` which contain the `controlConsumedEventGroups` and also *subscribe* to the `controlConsumedEventGroups`.](RS_SYST_00059)

[TPS_SYST_03029]{DRAFT} Initial *offer* for a provided service instance with *serviceControl* [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` and the *subscribe* to ALL referenced `controlConsumedEventGroups` was successful, then the translation software component shall *offer* the respective translated service instance.](RS_SYST_00059)

[TPS_SYST_03030]{DRAFT} **Stop offer for a provided service instance with `serviceControl`** [For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` and the *subscription* of at least ONE `controlConsumedEventGroup` is not available, then the translation software component shall *stop offer* the respective translated service instance.](RS_SYST_00059)

6.15.4 Translation behavior

There are two possible ways to define behavioral aspects for the signal-service-translation use-case:

- COM-Stack
- Translation Application Software Component

6.15.4.1 COM-Stack translation behavior

The Classic platform COM-Stack can be configured to have an own periodic behavior for periodic sending and reception/time-out monitoring. In this case the existing COM-Stack definition of behavior is used.

Example features are:

- `TransmissionModeTiming.cyclicTiming`
- `TransmissionModeTiming.eventControlledTiming`
- `IPduTiming.minimumDelay`
- `ISignalPort.timeout`

Thus it is possible to register the Translation Application Software Component to be notified when data arrives and then perform the translation operation solely driven by the notifications from the COM-Stack.

There are use-cases where the COM-Stack can not be used to perform the specific behavioral aspects because the Transformers are required to have access to the raw received and sent data. In these cases the usage of the Translation Application Software Component behavior is required.

6.15.4.2 Translation Application Software Component translation behavior

If the Translation Application Software Component shall implement specific behavioral aspect these have to be defined at the definition of the `CompositionSwComponentType` which has the signal-service-translation input and output `PortPrototypes` as well as the `PassThroughSwConnectors`.

For the periodic behavior the definition of `SenderComSpec.dataUpdatePeriod` and `ReceiverComSpec.dataUpdatePeriod` define the expected periods for data update and check.

[TPS_SYST_03042]{DRAFT} Periodic call in case of `ReceiverComSpec.dataUpdatePeriod` [If the signal-service-translation `CompositionSwComponentType` has a `RPortPrototype` defined with a `ReceiverComSpec.dataUpdatePeriod` defined then

the signal-service-translation software component implementation shall periodically call the respective Rte reception API with the defined period.](*RS_SYST_00059*)

[TPS_SYST_03043]{DRAFT} Periodic call in case of `SenderComSpec.dataUpdatePeriod` [If the signal-service-translation `CompositionSwComponentType` has a `PPortPrototype` defined with a `SenderComSpec.dataUpdatePeriod` defined then

the signal-service-translation software component implementation shall periodically call the respective Rte sending API with the defined period.](*RS_SYST_00059*)

Data filtering:

If there is a `filter` defined at the `NonqueuedReceiverComSpec` then the evaluation of this `DataFilter` is performed in the COM-Stack. Thus the COM-Stack filtering usually can not be applied when there are transformers involved because the state machines of E2E transformers need to receive every message.

If a data filtering shall be applied *after* the data transformation inside the signal-service-translation software component then there is the possibility to define a `DataFilter` at the `SignalServiceTranslationElementProps` in the role `filter`.

[TPS_SYST_03051]{DRAFT} Data filter inside the signal-service-translation software component [If there is a `SignalServiceTranslationElementProps.filter` defined this filtering shall be implemented inside the signal-service-translation software component.](*RS_SYST_00059*)

6.15.5 End-to-End considerations

In case there is an E2E header attached and/or a secure communication defined the translation needs to break the transport chain and re-calculate the E2E measures (CRC/MAC). The CRC/MAC is calculated over the uint8 array representation of the payload data. Since the Signal Service Translation changes the data layout, also the uint8 array representation of the payload data changes resulting in a different CRC/MAC value.

6.15.5.1 Safety

Due to the architectural approach of the Signal Service Translation the safety aspects can be configured using existing mechanisms. Thus it is possible to define dedicated safety end-to-end profiles for both the signal-based part, and the service-based part.

The translation software component then links the two parts together and handles the behavioral aspects of the translation.

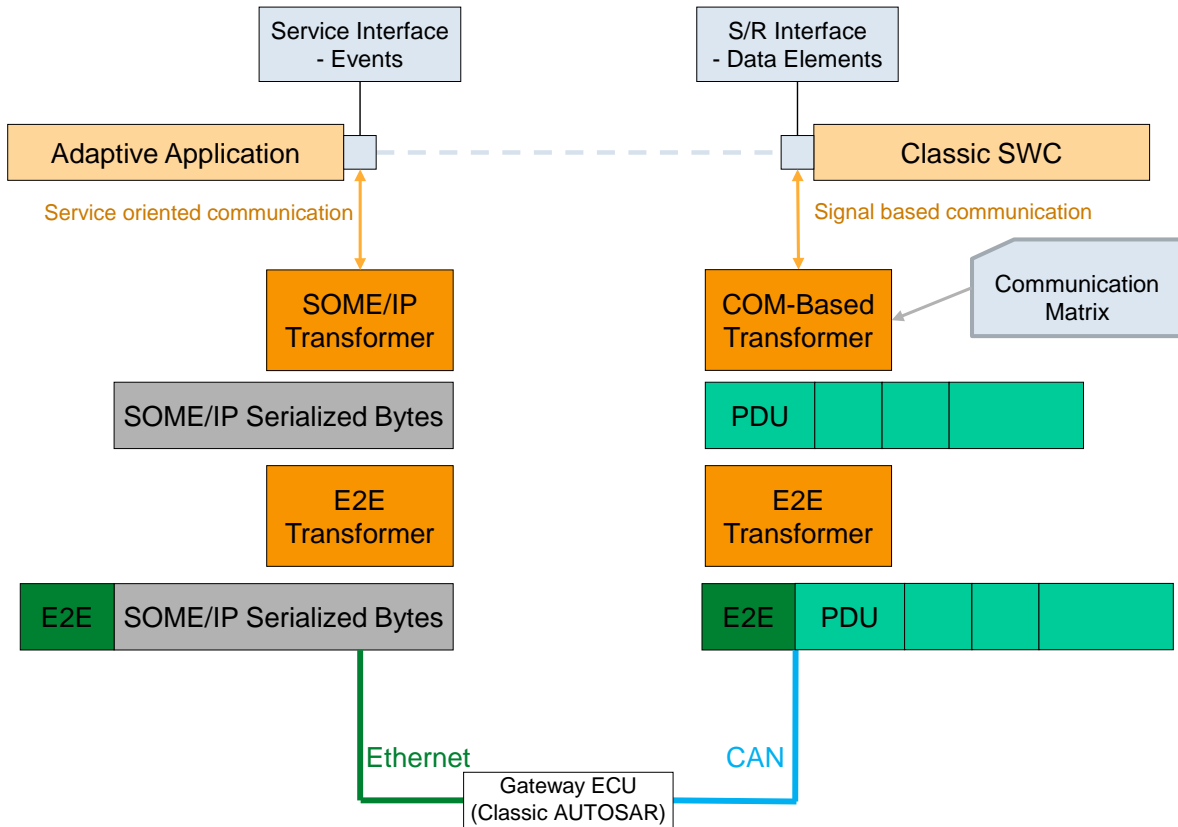


Figure 6.98: Signal Service Translation and Safety

The attribute `SignalServiceTranslationEventProps.safeTranslation` is used to explicitly require that both ends of the translation shall be configured in a safe transport way and that the translation software component shall handle the translation activity in a end-to-end preserving way.

[TPS_SYST_03044]{DRAFT} Handling of safe signal-service-translation in one software component [It is required that the signal-service-translation (and vice versa) of one Service/SignalGroup pair which are mapped to each other, shall be handled in one software component to also cover a closed mapping from one E2E profile to another, if necessary. The signal-service-translation of different (independent) Services/SignalGroups may be handled by different software component.]([RS_SYST_00059](#))

[TPS_SYST_03036]{DRAFT} PortAPIOption for safeTranslation RPortPrototype [If `SignalServiceTranslationEventProps.safeTranslation` is set to `true` then a `PortAPIOption` referring to the `RPortPrototype` shall exist and the `PortAPIOption.errorHandling` attribute shall be set to `transformer-ErrorHandling`.]([RS_SYST_00059](#))

[TPS_SYST_03037]{DRAFT} **PortAPIOption for safeTranslation PPortPrototype** [If `SignalServiceTranslationEventProps.safeTranslation` is set to `true` then a `PortAPIOption` referring to the `PPortPrototype` shall exist and the `PortAPIOption.transformerStatusForwarding` attribute shall be set to `transformerStatusForwarding`.] ([RS_SYST_00059](#))

[TPS_SYST_03045]{DRAFT} **Support for safe signal-service-translation and service-signal-translation** [The translation of E2E protected data shall be supported in both directions, signal-service-translation and service-signal-translation.] ([RS_SYST_00059](#))

[TPS_SYST_03046]{DRAFT} **Support for safe signal-service-translation with same or different E2E profiles** [The translation of E2E protected data shall support the occurrence of

- the same E2E profile on both sides of the communication and
- different E2E profiles on each side of the communication.

] ([RS_SYST_00059](#))

[TPS_SYST_03047]{DRAFT} **1:n mapping for E2E protected data** [It shall be possible to map the same E2E protected source data to several E2E protected target data (1:n).] ([RS_SYST_00059](#))

[TPS_SYST_03048]{DRAFT} **E2E protected target out of E2E protected sources** [The content of one E2E protected target shall only be composed out of data from E2E protected sources.] ([RS_SYST_00059](#))

The rationale for [TPS_SYST_03048] is to support the use-case where target data shall be E2E protected and is composed from several sources.

[TPS_SYST_03049]{DRAFT} **No translation of not OK E2E protected composed data** [If a E2E protected source data is mapped into a composed E2E protected target data and

if the E2E-Check for the source data returns any E2E error (not `E_OK`) then this source data shall not be forwarded to the respective target data and (if applicable) shall not trigger the transmission of the target.] ([RS_SYST_00059](#))

If source data is not verified as `E_OK` it is not translated into a composed target. If the translated E2E protected data comes from several sources there may occur correlation and synchronicity issues during translation.

[TPS_SYST_03031]{DRAFT} **Sufficient ASIL level of translation software component** [If the `SignalServiceTranslationEventProps.safeTranslation` equals true then the implementation of the translation software component shall fulfill a sufficient ASIL.] ([RS_SYST_00059](#))

[constr_3548]{DRAFT} **EndToEnd profile for both ends of safeTranslation** [If the `SignalServiceTranslationEventProps.safeTranslation` equals true then both, the signal-based payload as well as the service-oriented payload shall have an EndToEnd profile defined.] ()

[TPS_SYST_03032]{DRAFT} Data transmission in case of *E_OK* safe signal reception [Signal Service Translation shall check the end-to-end status of every received payload. If the safety transformer returns *E_OK* for the received payload then the data shall be forwarded to the respective sending of the translation software component.] ([RS_SYST_00059](#))

Error handling:

[TPS_SYST_03033]{DRAFT} No data transmission in case of reception timeout [If no message is received within the specified message cycle time (timeout is detected), then no data shall be translated to the respective sending of the translation software component.] ([RS_SYST_00059](#))

[TPS_SYST_03034]{DRAFT} Handling safe signal reception [The behavior of transformer status forwarding is defined in table 6.287.] ([RS_SYST_00059](#))

Source error	Forwarding status	Comment
E_OK	E_OK	–
E_SAFETY_VALID_REP	E_SAFETY_INVALID_REP	–
E_SAFETY_VALID_SEQ	E_SAFETY_INVALID_SEQ	–
E_SAFETY_VALID_ERR	E_SAFETY_INVALID_CRC	–
E_SAFETY_VALID_NND	NO TRANSLATION	No data received, thus no translation
E_SAFETY_NODATA_OK	E_OK	Statemachine is in state <i>No Data</i> , but received data is ok
E_SAFETY_NODATA_REP	E_SAFETY_INVALID_REP	Statemachine is in state <i>No Data</i> , but received data has repeated counter
E_SAFETY_NODATA_SEQ	E_SAFETY_INVALID_SEQ	Statemachine is in state <i>No Data</i> , but received data has wrong sequence counter
E_SAFETY_NODATA_ERR	E_SAFETY_INVALID_CRC	Statemachine is in state <i>No Data</i> , but received data has CRC Error
E_SAFETY_NODATA_NND	NO TRANSLATION	–
E_SAFETY_INIT_OK	E_OK	–
E_SAFETY_INIT_REP	E_SAFETY_INVALID_REP	–
E_SAFETY_INIT_SEQ	E_SAFETY_INVALID_SEQ	–
E_SAFETY_INIT_ERR	E_SAFETY_INVALID_CRC	–
E_SAFETY_INIT_NND	NO TRANSLATION	No data received, thus no translation
E_SAFETY_INVALID_OK	E_OK	Statemachine is in status <i>invalid</i> , but data itself is valid, thus no error replication
E_SAFETY_INVALID_REP	E_SAFETY_INVALID_REP	–
E_SAFETY_INVALID_SEQ	E_SAFETY_INVALID_SEQ	–
E_SAFETY_INVALID_ERR	E_SAFETY_INVALID_CRC	–
E_SAFETY_INVALID_NND	NO TRANSLATION	No data received, thus no translation
E_SAFETY_SOFT_RUNTIMEERROR	NO TRANSLATION	–
E_SAFETY_HARD_RUNTIMEERROR	NO TRANSLATION	–

Table 6.287: Error to Forwarding status mapping

6.15.5.2 Security

The security aspects are handled in the communication stack of Classic platform. There are two technologies for secure communication available:

- Secure Onboard Communication (SecOC) [30]
- Transport Layer Security (TLS) [31]

Which security technology is used for a translated service instance is up to the communication design. The security settings have to match between the providers and the consumers on the network.

As on Classic platform the translation happens on the application software component level it is possible to have equal, similar, or different security technologies configured on signal and service level (which is part of the COM-Stack configuration).

It is for instance well possible to have signals coming from a CanFD network secured with SecOC and translated into a service which is secured using TLS on Ethernet. But also using SecOC on Ethernet is possible.

In any case, if secure communication is involved together with signal-service-translation the translation needs to break the transport chain and re-secure the payload (e.g. re-calculate the MAC for the newly serialized payload).

The attribute `SignalServiceTranslationEventProps.secureTranslation` is used to explicitly require that both ends of the translation shall be configured in a secure transport way and that the translation software component shall handle the translation activity in a security-preserving way.

[constr_3549]{DRAFT} Secure payload for both ends in case of `secureTranslation` [If the `SignalServiceTranslationEventProps.secureTranslation` equals true then both, the signal-based payload as well as the service-oriented payload shall have a secure communication defined.]()

7 Data Transformation

7.1 Outline

The transmission of data over a communication bus requires some effort to convey the information about the nature of the transmitted data from the sender to the receiver. Both sides need to agree on this part or else the communication will fail.

This aspect is complicated by the fact that in most cases it is uncommon to transmit information in an atomic manner piece by piece. For the sake of properly utilizing the available communication resources, pieces of data that may or may not have any semantic relationship with each other are packed into a single transmission unit.

In this case, the receiver does not only have to be informed about the nature of the individual pieces of information but also about the packing of these pieces into the transmission unit.

There are different approaches of how this goal can be achieved, these are described in the following sub-chapters.

7.1.1 Configuration of the Communication Layout

Use a configurable software package on both the sender and the receiver side that can adapt to virtually any possible packing of data. In this case the packing shall be described in machine-readable form on a very detailed level in order to allow for the communication software to adapt to it.

For the sake of this argument, it doesn't really matter whether the adaption to the configuration is done at run-time or whether the configuration ends up in dedicated source code. The point is that the very detailed machine-readable configuration description is required to exist.

This approach used to be one of the pillars of the AUTOSAR standard as it entitled the players in the business with a maximum amount of flexibility and especially the OEMs are able to develop specific patterns for the design of their communication matrices that can, despite the diversity, be expressed with this approach.

This approach also facilitates the monitoring of transmission during development and deployment of the automotive software because monitoring tools can use the same configuration information to set themselves up for the task. This aspect is very important for debugging and quality assurance.

The downside, however, is that the act of laying out pieces of information in a limited number of transmission units becomes cumbersome and time-consuming. This effect becomes even more prominent with the advent of more advanced communication technologies that allow for a much bigger payload in single transmission units.

7.1.2 Data Transformation by Software

Don't care about the individual layout of information on the bus and let a piece of software take care of marshaling data onto the communication bus on the sender side and the reverse process on the receiver side.

This approach gains attractiveness in an environment where large and complicated pieces of information need to be transmitted.

Of course, in order to make this approach work it is necessary to standardize the behavior of the marshaling software to the necessary extent such that sender and receiver agree on how data needs to be processed.

With this approach, the amount of configuration can be reduced dramatically at the potential expense of efficiency and code size.

But this is not the end of the story as the idea of letting software take care of data "manipulation" can **following pretty much the same pattern** be utilized for further use cases:

End-to-end Protection Data is wrapped into a harness of meta-data that allows for checking data integrity at the receiver side.

Data Security Data is cryptographically processed such that it shall become impossible for unauthorized parties to intercept the communication process.

In other words, the approach is not limited to marshaling of data but can in the same way also be used for an array of other useful data transformations. This is why the terminology in this regard is not limited to the marshaling but to data transformation in general, hence the term **Data Transformer** is coined.

`Data Transformers` can be chained such that, on the sender side, one `Data Transformer` picks up the result of the transformation of another `Data Transformer` and applies a specific transformation to the already processed data.

The receiver then is required to apply the `Data Transformers` in reverse order in order to finally yield the actual data and provide it to the consumer (e.g. an [ApplicationSwComponentType](#)).

A basic principle of the `Data Transformer` approach, however, is that the `Data Transformer` is only responsible for the actual data transformation but **not** concerned about the communication of data. This can be taken care of by other software modules.

In total, the second approach provides a sufficient level of utility that it becomes part of the AUTOSAR standard. This chapter lays out the details of how `Data Transformers` can be used in the context of this document.

Further information can also be found in the SWS RTE [32].

7.2 Use Cases

This chapter describes Transformer use cases that are supported by AUTOSAR.

7.2.1 Transmission of large composite data types over networks with large PDUs (e.g Ethernet)

With a serializing transformer, it is not necessary any more to map the atomic sub-elements of composite data types to individual signals in the RTE. The sending application SWC sends the composite data element using Sender/Receiver communication and hands the data over to the RTE. Then the complex data get transformed to a linear byte array and handed over to Com which sends the data to the receiving ECU. There, the Com stack receives the serialized data and notifies the RTE. The Rte reads the data and calls the deserializing transformer. The deserializing transformer transforms them back into the composite data element and gives the result to the RTE. The receiving SWC can now read the data and access it in the same form the sending SWC has sent them.

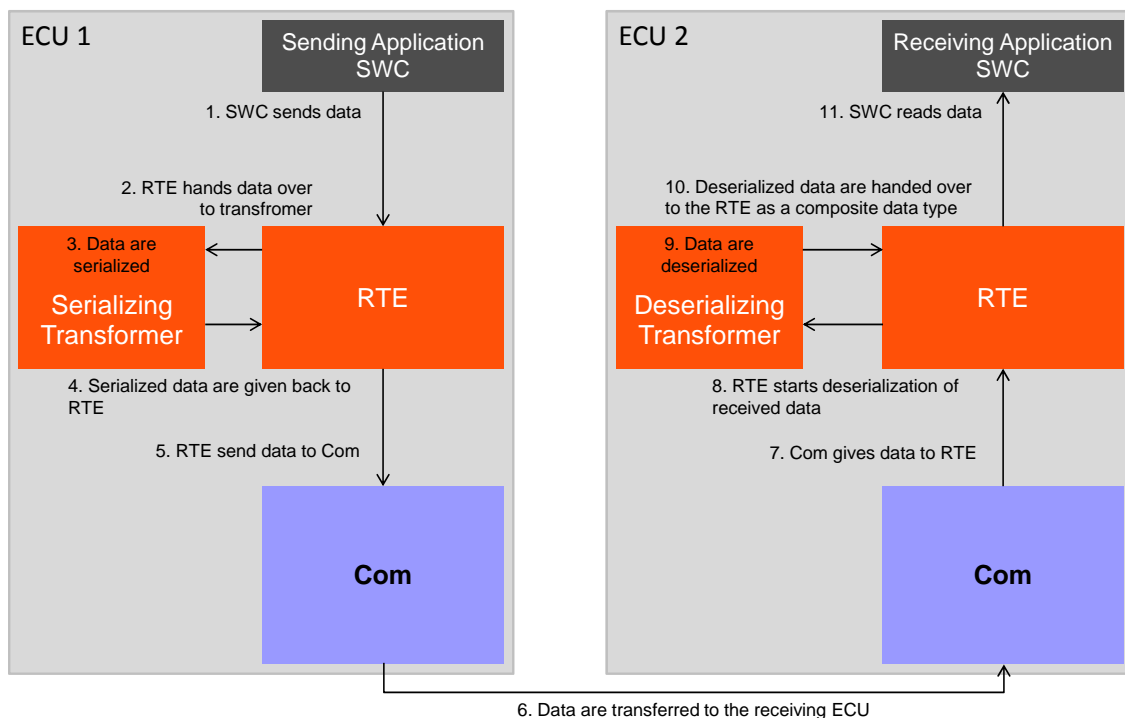


Figure 7.1: Transformer Use Case: Transmission of large composite data types over networks with large PDUs (e.g Ethernet)

7.2.2 Support of transmission from one sender to multiple receivers with Signal Fan-out

If a signal fan-out is configured in the System Description, the RTE has to hand over the data which should be transmitted multiple times to the Com stack. This is the case if multiple *ISignals* reference the same *SystemSignal* in the System Description.

For each *ISignal* the following steps have to be performed individually:

- transform the data
- hand it over to COM

Every receiver has to deserialize the *ISignal* in its transformer independently.

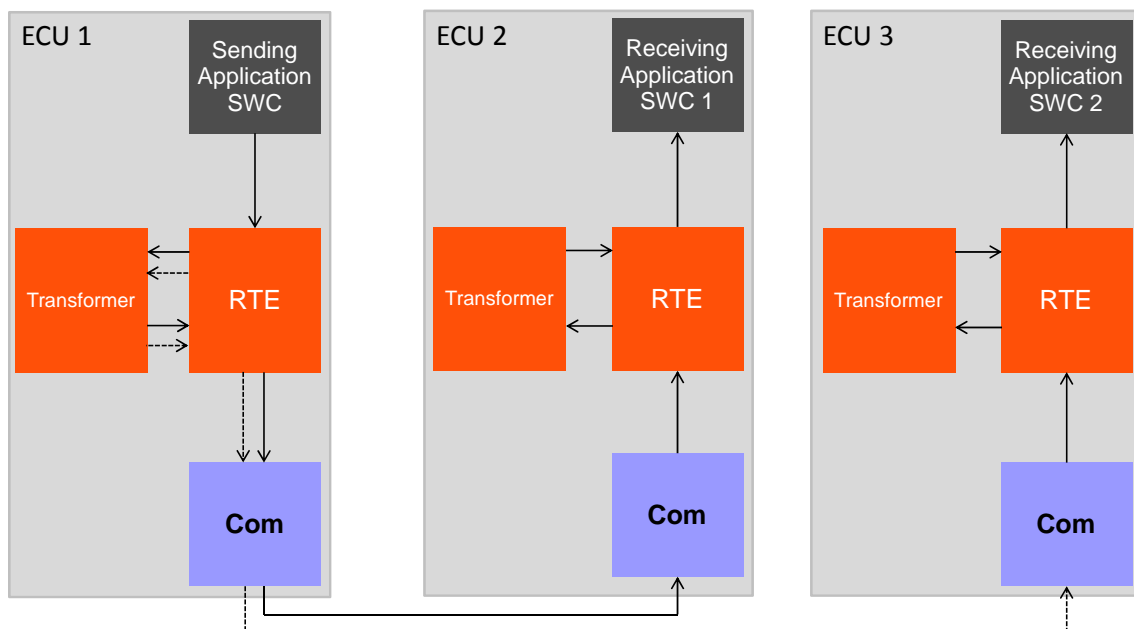


Figure 7.2: Transformer Use Case: RTE Fanout

7.2.3 Support of transmission from one sender to multiple receivers with PDU Fan-out

The transformation of inter-ECU Sender/Receiver communication should also work together with configurations that include *Pdu* fan-outs inside the COM stack (*PduR* fan-out). This is the case if multiple *PduTriggerings* reference the same *Pdu* in the System Description. In that scenario the data are sent by the sending application SWC to the RTE and transformed by the data transformer which is called by the RTE. Then the RTE hands the data over the Com. This happens only once. Due to the *Pdu* fan-out, the *PduR* sends the data multiple times to the Bus Interfaces using different *Pdus*.

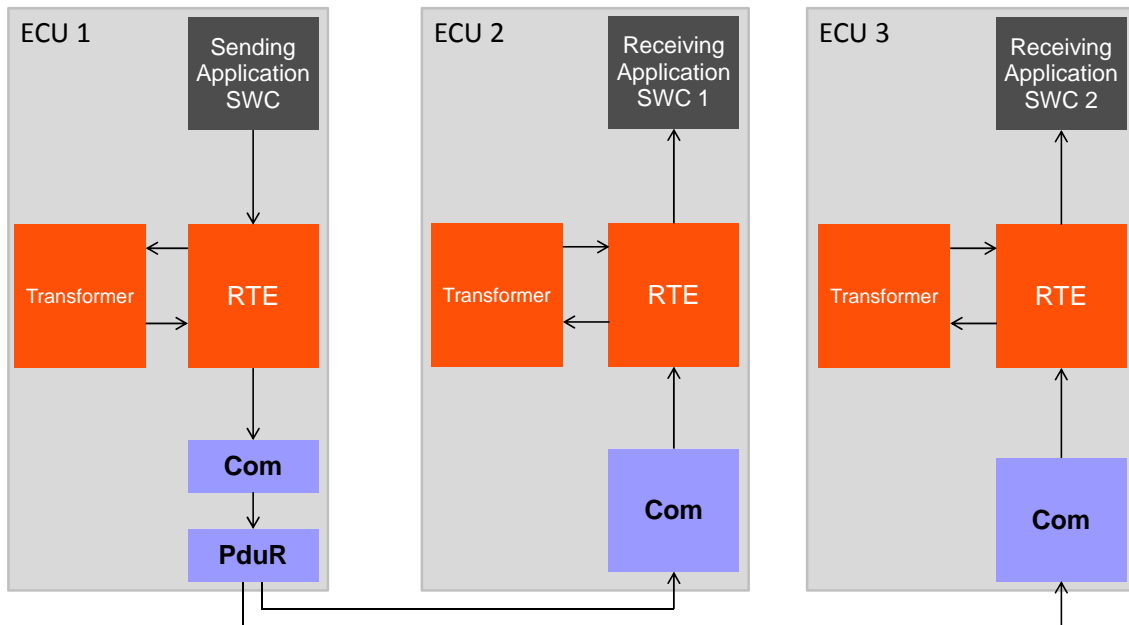


Figure 7.3: Transformer Use Case: PduR Fanout

7.2.4 Transformer Chaining

It is possible to chain multiple transformers. The output of one transformer then will be the input of the next transformer in the chain. Transformer for serialization data, for encrypting, digitally signing or compressing data can be implemented and used together. Such architecture could be used to assemble a system, where you can flexibly add functionality like compression or encryption to a serialized stream. In AUTOSAR the E2E-protection is implemented by an additional serializer which is appended to the chain.

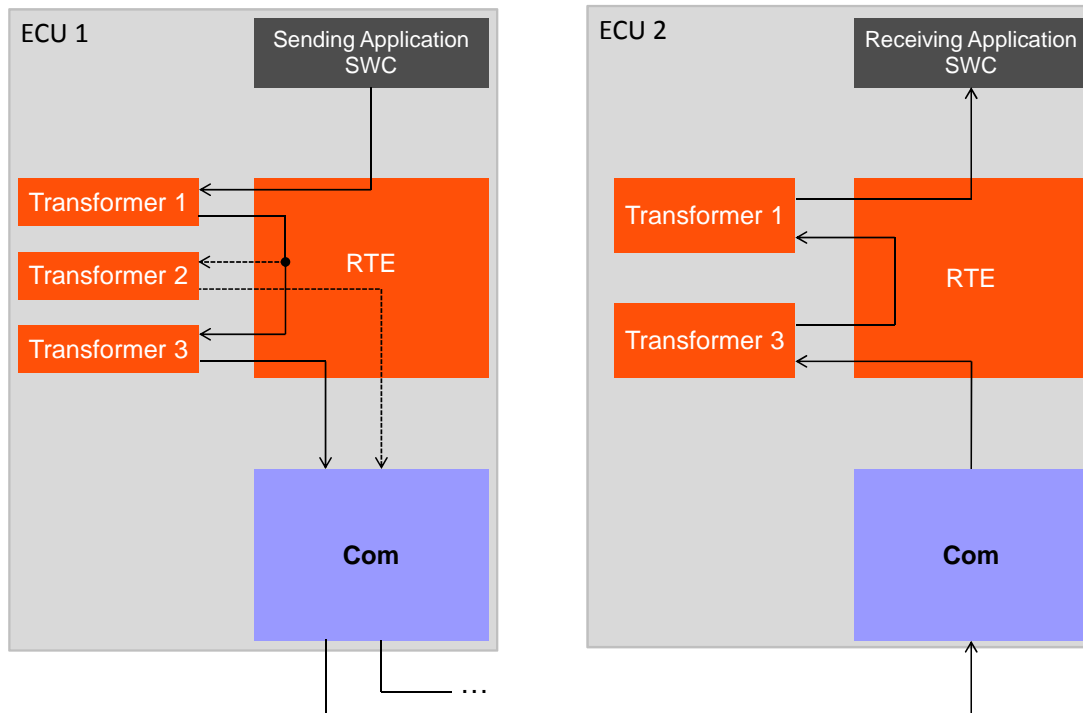


Figure 7.4: Transformer Use Case: Transformer Chain

7.2.5 Signal Group Based interaction of the transformer with the Com module

An initial transformer (serializer) performs the serialization according to the `ISignalToIPduMapping` from the system description. For each application data element the corresponding mapping to an `ISignalIPdu` position is respected. After the transformation chain is processed the serialized data is provided to the Com module. The Com module can have a signal based transmission mode selection defined and determines the respective transmission mode to be applied.

7.3 Transformer configuration

As a transformer provides well defined function signatures per each communication relation (`ISignal` based), which is marked for transformation, the function signature is NOT dependent from the transformation technology used, but only from the transmitted data elements (Client/Server operation signature or Sender/Receiver interface signature). The output of a transformer will be always a linear byte array.

Configuration of data transformation consists of three parts:

1. definition of the transformer chains with their transformers
2. configuration which communication is subject to transformation

3. configuration of the transformer properties for the transformed communication

The configuration of single transformers and whole transformer chains is shown in figure 7.5.

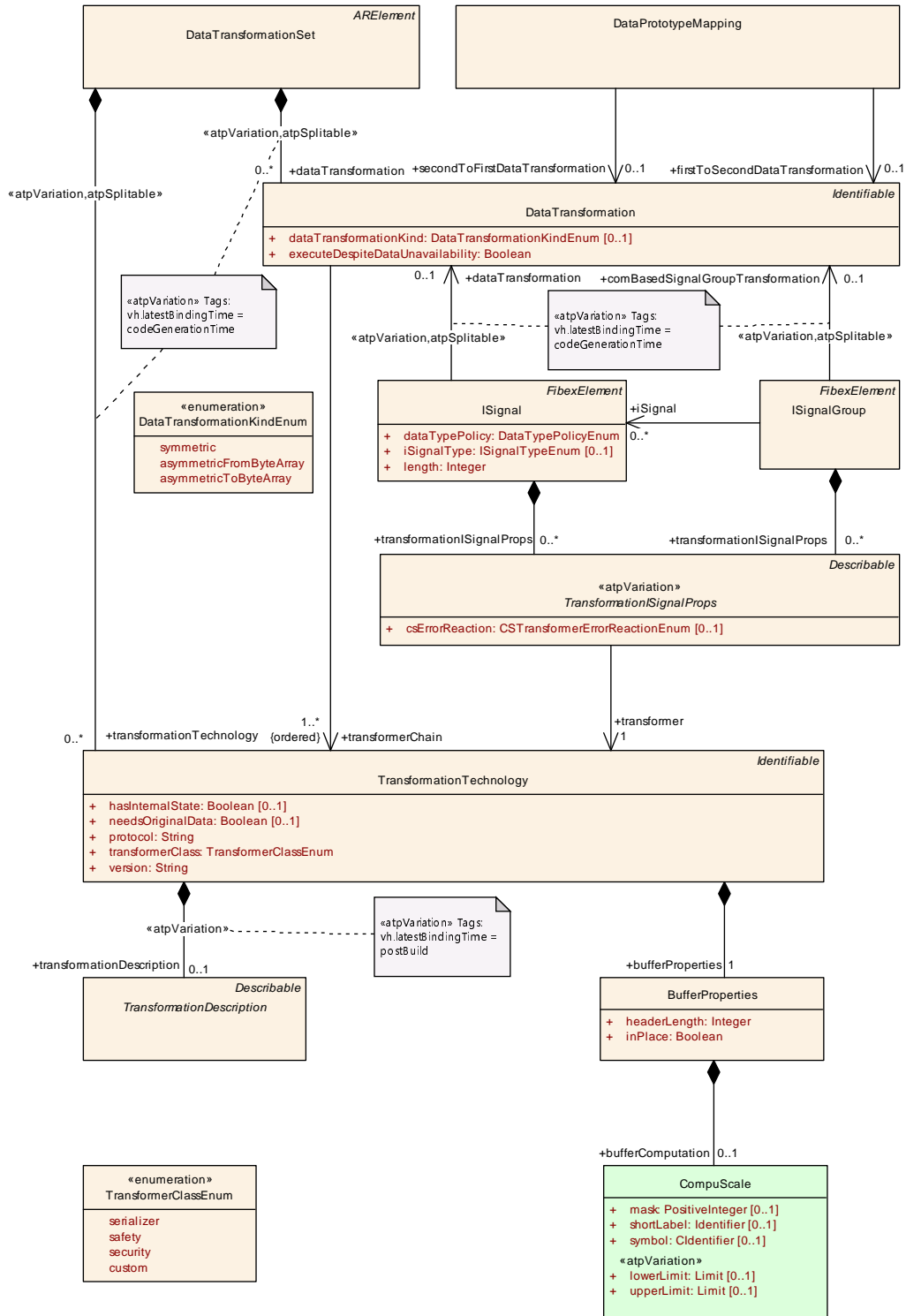


Figure 7.5: Configuration of transformers and transformer chains

The [DataTransformationSet](#) acts as a central container for the configuration of data transformation.

Class	DataTransformationSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This element is the system wide container of DataTransformations which represent transformer chains. Tags: atp.recommendedPackage=DataTransformationSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
data Transformation	DataTransformation	*	aggr	This container consists of all transformer chains which can be used for transformation of data communication. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
transformation Technology	TransformationTechnology	*	aggr	Transformer that is used in a transformer chain for transformation of data communication. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime

Table 7.1: DataTransformationSet

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			
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)	TransformationTechnology	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 7.2: DataTransformation

[TPS_SYST_02030] The [DataTransformationSet](#) contains all transformer chains [The [DataTransformationSet](#) contains transformer chains represented by [DataTransformation](#) elements.] ([RS_SYST_00050](#))

For each transformer chain it can be decided via the attribute [executeDespiteDataUnavailability](#) whether the RTE should try to execute the transformers of the transformer chain, even when no data are available as input. e.g. the queue is empty or there was an error in the COM stack. This is needed when no data are available but a transformer has to be executed anyway because it maintains an internal state which has to be updated to consider that data was expected but not available. This might be used in transformers which maintain an internal state. Of course the

specifications and implementations of all transformers in the chain have to be able to cope with execution without valid input data.

[constr_3208] executeDespiteDataUnavailability usage restriction [In the set of more than one `ISignal` which reference the same `SystemSignal` in the role `systemSignal`, there shall be no `ISignal` which references a `DataTransformation` where `executeDespiteDataUnavailability` is set to true.] ()

In other words: There shall be no transformer chain which "belong" to the same `SystemSignal` due to signal fan-in where the attribute `executeDespiteDataUnavailability` is set to true.

[TPS_SYST_02031] A transformer is represented by a TransformationTechnology [A transformer is represented by a `TransformationTechnology`.] (*RS_SYST_00050*)

Class	TransformationTechnology			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A TransformationTechnology is a transformer inside a transformer chain. Tags: xml.namePlural=TRANSFORMATION-TECHNOLOGIES			
Base	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
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 Transformer Description. Stereotypes: atpVariation Tags: 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 7.3: TransformationTechnology

Enumeration	TransformerClassEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Specifies the transformer class of a transformer.
Literal	Description
custom	The transformer is a custom transformer. Tags: atp.EnumerationLiteralIndex=0
safety	The transformer is a safety transformer. Tags: atp.EnumerationLiteralIndex=1





Enumeration	TransformerClassEnum
security	The transformer is a security transformer. Tags: atp.EnumerationLiteralIndex=2
serializer	The transformer is a serializing transformer. Tags: atp.EnumerationLiteralIndex=3

Table 7.4: TransformerClassEnum

[constr_3265] TransformationTechnology.hasInternalState setting for an E2E transformer [The value of `hasInternalState` shall be set to true for a `TransformationTechnology` with `transformerClass` set to `safety`.]()

[constr_3266] TransformationTechnology.hasInternalState setting for a SOME/IP Transformer [The value of `hasInternalState` shall be set to true for a SOME/IP Transformer if `SOMEIPTransformationISignalProps.sessionHandlingSR` for the `ISignal` is set to active.]()

[TPS_SYST_02032] Transformer chains are ordered list of transformers [A transformer chain consists of an ordered list of `TransformationTechnologies` (transformers).](*RS_SYST_00050*)

[constr_3121] The length of transformer chains is limited to 255 transformers [The maximum number of `DataTransformation.transformerChain` references in the context of one `DataTransformation` shall be limited to 255.]()

[constr_3122] At most one transformer of each transformer class inside a transformer chain [If the value of a `transformerClass` of a `TransformationTechnology` referenced by a `DataTransformation` does not equal `custom`, it shall be different from all other `transformerClass` values of `TransformationTechnologies` referenced by the same `DataTransformation`.]()

Only for `custom` transformers it is possible to specify more than one transformer of the same class in the same transformer chain. For all other transformer classes, at most one transformer of a transformer class is allowed to exist in the same transformer chain.

[constr_3123] Serializer transformer shall be the first in a chain [A serializer transformer (`TransformationTechnology` with attribute `transformerClass` set to `serializer`) shall be the first transformer in a transformer chain.]()

[TPS_SYST_02033] Order of the transformerChain references in the configuration represents the order on the sending side [The order of `DataTransformation.transformerChain` references in the context of one `DataTransformation` represents the transformation order on the sending side.](*RS_SYST_00050*)

[TPS_SYST_02034] Order of the transformers on the receiving side is the reverse of the sending side [The order of the transformers on the receiving side of the data shall be the inverse order of the order of the sending side.](*RS_SYST_00050*)

[TPS_SYST_02035] protocol contains the human readable protocol identifier [The attribute `protocol` of a `TransformationTechnology` contains the protocol name as a String which this transformer implements.] (*RS_SYST_00050*) This attribute is used to distinguish transformers in a human readable way.

[TPS_SYST_02036] version contains the version of the protocol [The attribute `version` of a `TransformationTechnology` contains the version of the protocol as a String implemented by this transformer.] (*RS_SYST_00050*) This attribute is used to distinguish transformers.

[TPS_SYST_02037] The attribute needsOriginalData configures a transformer's access to the original data [The attribute `needsOriginalData` of a `TransformationTechnology` specifies whether transformer needs access to the original data.] (*RS_SYST_00050*)

If it is set to true, the transformer will gain access to the original data. If it is set to false, the transformer will not gain access to the original data.

[constr_3124] Applicability of needsOriginalData [The attribute `needsOriginalData` of a `TransformationTechnology` shall only be used for the non-first transformers in the transformer chain.] ()

This will only influence the signatures of the transformer on the sender or client side, not on the receiver or server side of a communication.

[TPS_SYST_02038] Specification of transformer class [The transformer class to which this transformer belongs to is specified in the attribute `transformerClass` of a `TransformationTechnology`] (*RS_SYST_00050*)

[TPS_SYST_02039] Specification of transformer specific properties [Further transformer specific properties can be stated inside the `TransformationDescription` in the role `transformationDescription` of a `TransformationTechnology`] (*RS_SYST_00050*)

Note:

This is an abstract class without any specified content. If AUTOSAR specifies a transformer and this transformer need configuration possibilities, this class can be inherited to hold those as some kind of container.

[TPS_SYST_02040] Specification of transformer buffer handling [The `BufferProperties` in the role `bufferProperties` of a `TransformationTechnology` specify the buffer handling which shall be executed by the RTE for this transformer.] (*RS_SYST_00050*)

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

[TPS_SYST_02041] In-place buffer handling of transformers [The attribute `inPlace` of `BufferProperties` specifies whether the transformation happens in-place.] ([RS_SYST_00050](#))

[constr_3125] Value of attribute `inPlace` for the first transformer in a chain [The attribute `inPlace` shall be set to `false` if the `TransformationTechnology` of the `BufferProperties` is referenced as first reference in the ordered list of references `transformerChain` from a `DataTransformation`.] ()

[TPS_SYST_02042] Header length to be considered by transformers [The attribute `headerLength` of `BufferProperties` specifies the length of the header (in bits) which the transformer adds.] ([RS_SYST_00050](#))

[constr_3126] `headerLength` shall be less or equal output buffer size [The `headerLength` shall be less or equal of the worst case output buffer size which is specified in `bufferComputation` in `BufferProperties`.] ()

[constr_3364] `headerLength` shall be a multiple of 8 [The header length in bits specified by `headerLength` shall be a multiple of 8.] ()

[TPS_SYST_02043] Buffer computation of transformer [The `CompuScale` aggregated as `bufferComputation` in `BufferProperties` specifies the computation formula how the transformer changes the buffer length between input and output. This enables the calculation of a safe worst case output size.] ([RS_SYST_00050](#))

[constr_3314] `BufferProperties.bufferComputation` is mandatory [The `BufferProperties` that is aggregated by `TransformationTechnology` in the role `bufferProperties` shall always define the `bufferComputation`.] ()

[TPS_SYST_02044] Buffer computation of transformer [The `bufferComputation` in `BufferProperties` shall specify the formula:
 $OutputBufferLength = CompuScale(InputBufferLength)$, where `InputBufferLength` is computed according to [SWS_Rte_03867].] ([RS_SYST_00050](#))

The `CompuScale` in the role `bufferComputation` defines the list of polynomial factors. The first factor represents the power of zero. Please note that the `headerLength` does NOT separately affect the size of the output buffer.

The following series of XML fragments exemplifies how the `CompuScale` in the role `bufferComputation` can be used to describe the buffer needs of a transformer.

The following XML in Listing 7.1 shows the buffer properties of transformer which extends the input data length by adding a header with the length of 8 bytes and a tail of 4 bytes. Therefore the buffer needs to grow by 12 bytes.

$$OutputBufferSize = 8Bytes + 1 * InputDataLength + 4Bytes$$

Listing 7.1: Example for a `CompuScale` which specifies a buffer that grows by 12

```

<TRANSFORMATION-TECHNOLOGY>
  <SHORT-NAME>Transformer1</SHORT-NAME>
  <BUFFER-PROPERTIES>
    <BUFFER-COMPUTATION>
      <COMPU-RATIONAL-COEFFS>
        <COMPU-NUMERATOR>
          <V>12</V>
          <V>1</V>
        </COMPU-NUMERATOR>
        <COMPU-DENOMINATOR>
          <V>1</V>
        </COMPU-DENOMINATOR>
      </COMPU-RATIONAL-COEFFS>
    </BUFFER-COMPUTATION>
    <HEADER-LENGTH>64</HEADER-LENGTH>
    <IN-PLACE>>false</IN-PLACE>
  </BUFFER-PROPERTIES>
  <PROTOCOL>Serializer</PROTOCOL>
  <TRANSFORMER-CLASS>SERIALIZER</TRANSFORMER-CLASS>
  <VERSION>1.0.0</VERSION>
</TRANSFORMATION-TECHNOLOGY>
    
```

The following XML in Listing 7.2 shows the buffer properties of transformer which doubles the length of the data in the transformer and adds additional 8 bytes as header.

$$OutputBufferSize = 8Bytes + 2 * InputDataLength$$

Listing 7.2: Example for a `CompuScale` which specifies where the data size double and additionally grows by 8

```

<TRANSFORMATION-TECHNOLOGY>
  <SHORT-NAME>Transformer1</SHORT-NAME>
  <BUFFER-PROPERTIES>
    <BUFFER-COMPUTATION>
      <COMPU-RATIONAL-COEFFS>
        <COMPU-NUMERATOR>
          <V>8</V>
          <V>2</V>
        </COMPU-NUMERATOR>
        <COMPU-DENOMINATOR>
          <V>1</V>
        </COMPU-DENOMINATOR>
      </COMPU-RATIONAL-COEFFS>
    </BUFFER-COMPUTATION>
    <HEADER-LENGTH>8</HEADER-LENGTH>
    <IN-PLACE>>false</IN-PLACE>
  </BUFFER-PROPERTIES>
  <PROTOCOL>Serializer</PROTOCOL>
  <TRANSFORMER-CLASS>SERIALIZER</TRANSFORMER-CLASS>
  <VERSION>1.0.0</VERSION>
</TRANSFORMATION-TECHNOLOGY>
    
```

```

        </COMPU-DENOMINATOR>
        </COMPU-RATIONAL-COEFFS>
    </BUFFER-COMPUTATION>
    <HEADER-LENGTH>64</HEADER-LENGTH>
    <IN-PLACE>>false</IN-PLACE>
</BUFFER-PROPERTIES>
<PROTOCOL>Serializer</PROTOCOL>
<TRANSFORMER-CLASS>SERIALIZER</TRANSFORMER-CLASS>
<VERSION>1.0.0</VERSION>
</TRANSFORMATION-TECHNOLOGY>

```

Note: Listing 7.1 and Listing 7.2 deliberately leave out any `TransformationTechnology` specific attributes, since both listings are about `BufferProperties.bufferComputation` only.

[constr_3286] `ISignal.length` shall be consistent to transformer configuration [For `ISignals` that are used for transformed data, the value `ISignal.length` shall be greater or equal to the maximum possible size of the transformed data (including alignment). This size can be calculated by using the formulas specified in the `TransformationTechnology.bufferProperties.bufferComputation` of all `TransformationTechnologies` in the ordered list `DataTransformation.transformerChain` for the length that is determined from the mapped `VariableDataPrototype`.]()

The following examples are showing the calculation of the `length` of an `ISignal` that transports a `VariableDataPrototype` of `ImplementationDataType` of category `STRUCTURE` via a `SOME/IP` Transformer.

The example struct consists of five members:

Member1: `UINT16`

Member2: Struct with a `UINT16` length field and a one-dimensional variableSize Array with `UINT8` elements and `arraySize = 8`

Member3: `UINT32`

Member4: `UINT64`

Member5: Struct with a `UINT16` length field and a one-dimensional variableSize Array with `UINT8` elements and `arraySize = 8`

The `SOME/IP` Transformer takes the `InputData` and adds additional 8 bytes as header. In case of `SOME/IP` the signal based `SOMEIPTransformationISignalProps` and the `DataPrototype` based `SOMEIPTransformationProps` need to be considered as well for the calculation of the `ISignal.length` (see chapter 7.3.2.2 for more details). In our example the following `SOMEIPTransformationProps` settings are valid for the variableSize Array:

- `SOMEIPTransformationProps.alignment = 64`

All these settings lead to the `ISignal.length` of 368 bits as shown in figure Figure 7.6. A padding element is inserted after the first variable size array as described in

[PRS_SOMEIP_00611]. The second variable size array is the last element in the serialized data stream and therefore no padding element is inserted afterwards. The automatic padding in SOME/IP after variable size data is described in more detail in [33].

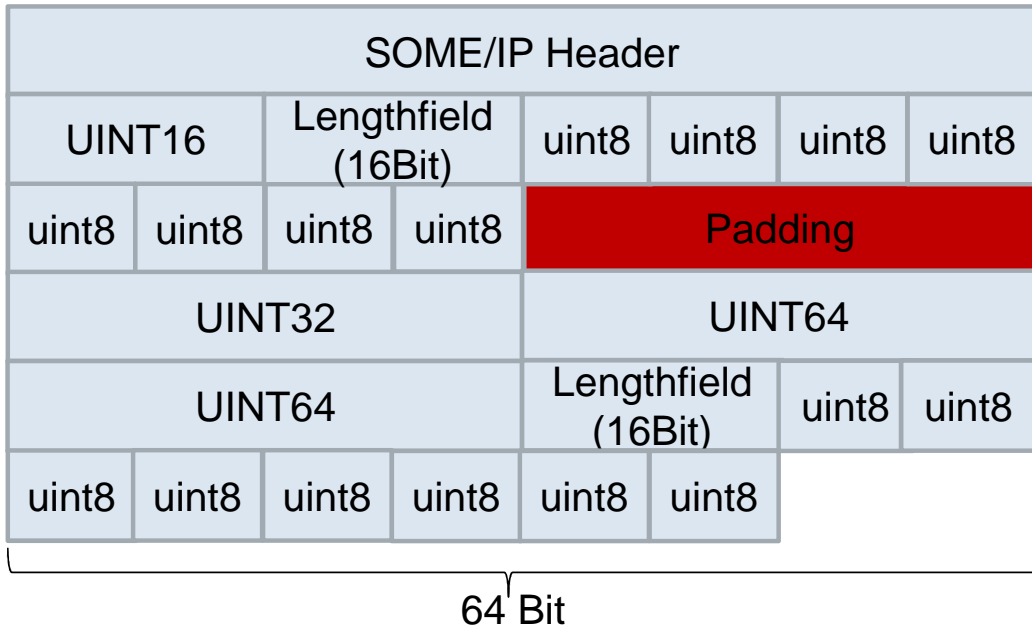


Figure 7.6: Example for calculation of the `ISignal.length`

Please note that the padding in the SOME/IP data stream depends on the actual number of elements that are transmitted in the variable data. Figure 7.7 shows an example where only three elements are transmitted in the first variable size array and therefore the padding is restricted to 1 byte.

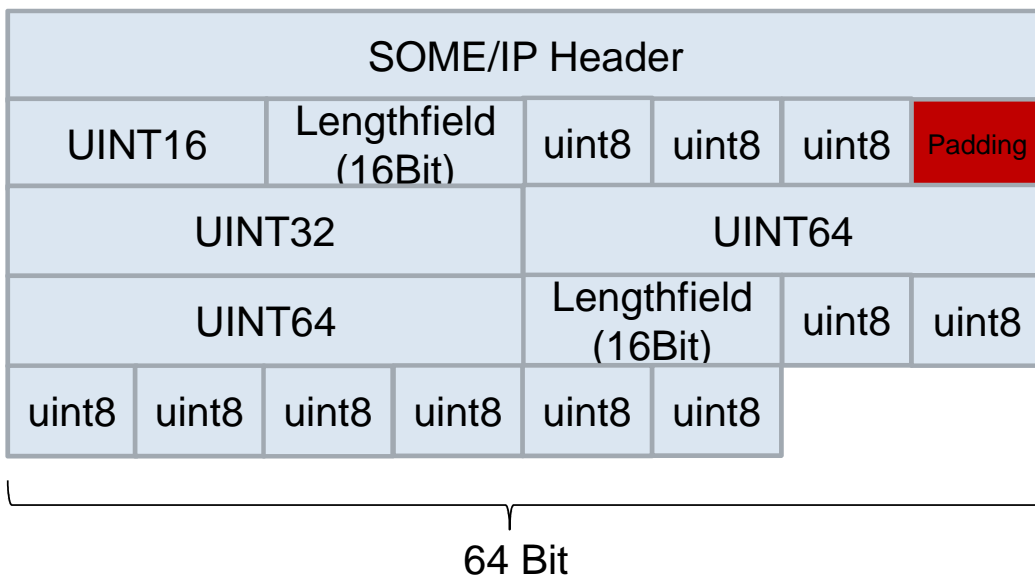


Figure 7.7: SOME/IP Padding Example

Transformer specific configuration can be done in the [TransformationDescription](#).

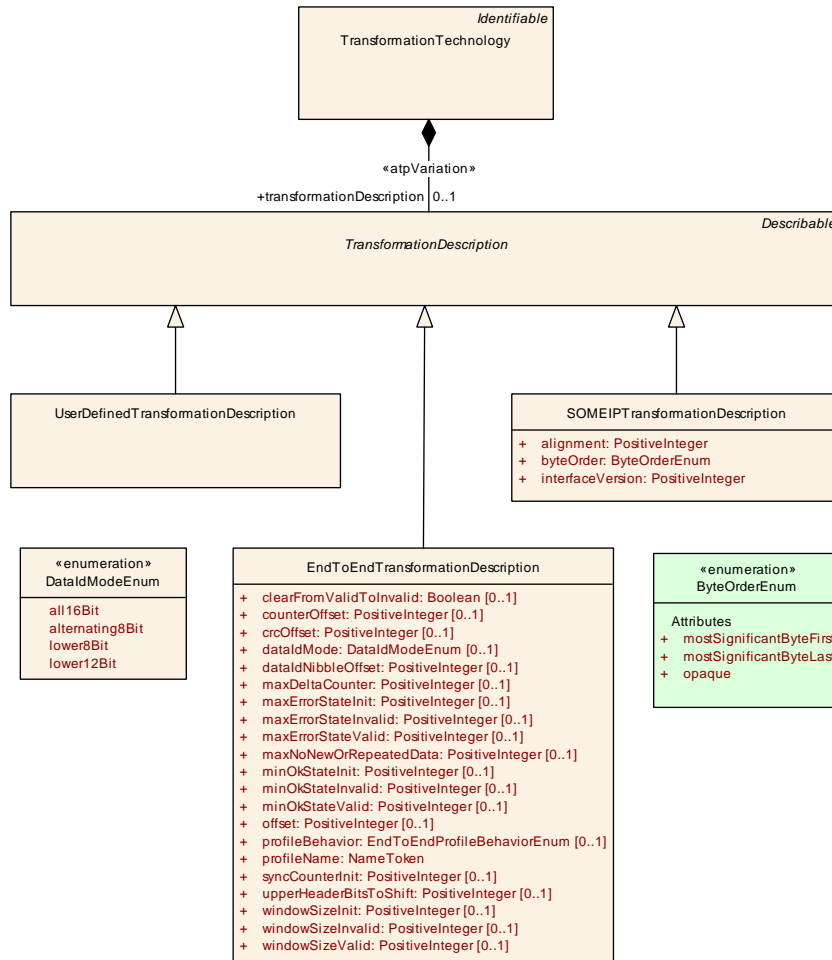


Figure 7.8: Configuration of transformers using [TransformationDescription](#)

Class	<i>TransformationDescription</i> (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 , UserDefinedTransformationDescription			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 7.6: TransformationDescription

Class	UserDefinedTransformationDescription			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The UserDefinedTransformationDescription is used to specify details and documentation for custom transformers.			
Base	<i>ARObject</i> , <i>Describable</i> , <i>TransformationDescription</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 7.7: UserDefinedTransformationDescription

[TPS_SYST_02045] SOME/IP Transformer configuration [SOME/IP Transformer shall be configured using *SOMEIPTransformationDescription*.] (*RS_SYST_00050*)

[TPS_SYST_02046] E2E Transformer configuration [E2E Transformer shall be configured using *EndToEndTransformationDescription*.] (*RS_SYST_00050*)

For details how to configure those transformers please see chapter 7.3.2 and chapter 7.3.4.

[TPS_SYST_02047] Custom transformer configuration [For custom transformers the specific configuration options shall be placed inside *UserDefinedTransformationDescription*.] (*RS_SYST_00050*)

To place the custom data in *UserDefinedTransformationDescription* the *AdminData* could be used for example.

The configuration in *TransformationDescription* is valid for the transformer (*TransformationTechnology*) and all associated *ISignals*. If *ISignal* specific configuration shall be realized which is only valid for the transformation of a specific *ISignal*, the *TransformationISignalProps* shall be used.

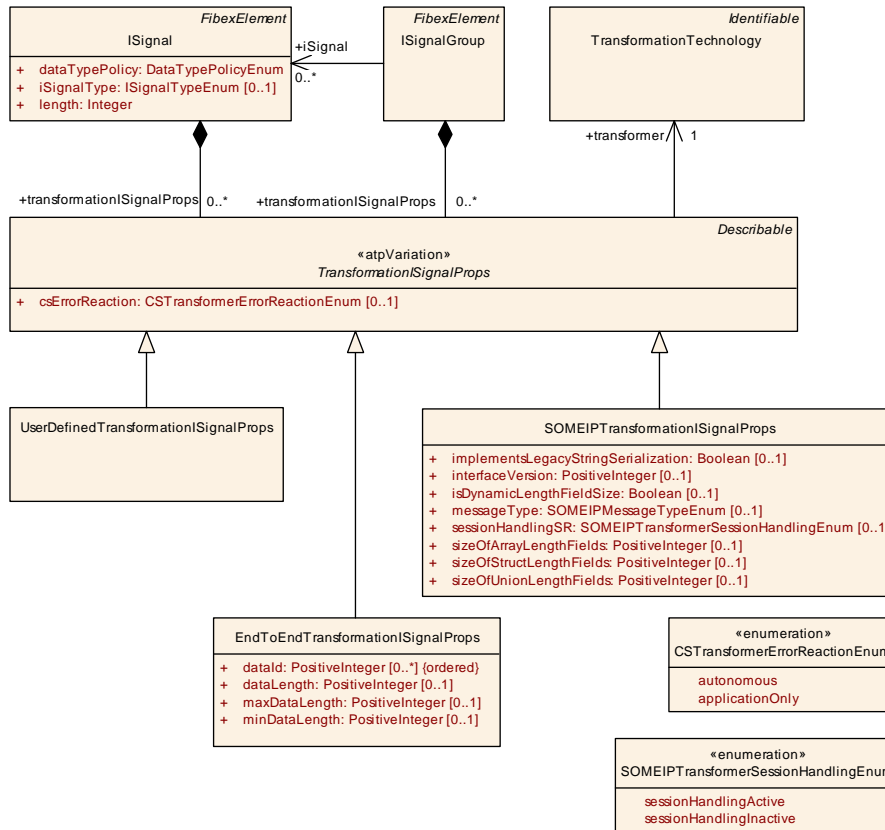


Figure 7.9: Configuration of transformers using TransformationISignalProps

Class	<<atpVariation>> TransformationISignalProps (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	TransformationISignalProps holds all the attributes for the different TransformationTechnologies that are ISignal specific. Tags: vh.latestBindingTime=postBuild			
Base	ARObject, Describable			
Subclasses	EndToEndTransformationISignalProps, SOMEIPTransformationISignalProps, UserDefinedTransformationISignalProps			
Attribute	Type	Mult.	Kind	Note
csErrorReaction	CSTransformerErrorReactionEnum	0..1	attr	Defines whether the transformer chain of client/server communication coordinates an autonomous error reaction together with the RTE or whether any error reaction is the responsibility of the application.
dataPrototype Transformation Props	DataPrototype TransformationProps	*	aggr	Fine granular modeling of TransformationProps on the level of DataPrototypes.
transformer	TransformationTechnology	1	ref	Reference to the TransformationTechnology description that contains transformer specific and ISignal independent configuration properties.

Table 7.8: TransformationISignalProps

Enumeration	CSTransformerErrorReactionEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Possible kinds of error reaction in case of a hard transformer error.
Literal	Description
applicationOnly	The application is responsible for any error reaction. No autonomous error reaction of RTE and transformer. Tags:atp.EnumerationLiteralIndex=0
autonomous	RTE and Transformer coordinate an autonomous error reaction on their own. Tags:atp.EnumerationLiteralIndex=1

Table 7.9: CSTransformerErrorReactionEnum

[TPS_SYST_02048] ISignal specific transformation configuration [If an `ISignal` references a `TransformationTechnology` in the role `dataTransformation` and this transformation shall be configured `ISignal` specific, the `ISignal` shall aggregate a `TransformationISignalProps` element.] (*RS_SYST_00050*)

[TPS_SYST_02049] Transformer specific TransformationISignalProps [The attribute `transformer` of `TransformationISignalProps` shall reference the `TransformationTechnology` in the transformer chain (`DataTransformation`) for which the `ISignal` specific configuration shall be given.] (*RS_SYST_00050*)

[constr_3213] TransformationISignalProps.csErrorReaction setting in case that the serializer transformerClass and Client/Server communication is used [In `TransformationISignalProps` the attribute `csErrorReaction` shall be set if the `TransformationISignalProps` specifies the details for a `TransformationTechnology` with `transformerClass` equal to `serializer` and the `ISignal` that aggregates the `TransformationISignalProps` transports a client/server communication.] ()

[constr_3214] TransformationISignalProps.csErrorReaction setting in case that a transformerClass different from serializer is used or the Client/Server communication is not used [In `TransformationISignalProps` the attribute `csErrorReaction` shall not be used if the `TransformationISignalProps` specifies the details for a `TransformationTechnology` with `transformerClass` not equal to `serializer` or the `ISignal` that aggregates the `TransformationISignalProps` does not transport a client/server communication.] ()

[TPS_SYST_02074] Precedence of transformer configuration settings [The same transformer configuration settings may exist in the `TransformationDescription`, `TransformationISignalProps` and `TransformationComSpecProps` elements. The following precedence is valid for such settings:

- `TransformationDescription`: configuration valid for several `ISignals` (in case the SOME/IP Transformer or Custom Transformer is used) or `ISignalGroups` (in case the ComBasedTransformer is used).
- `TransformationISignalProps`: defines the configuration options valid for a specific referenced `ISignal` or `ISignalGroup`. This settings override possible settings in the `TransformationDescription`.

- `TransformationComSpecProps`: defines the configuration settings valid for the port to which the `ReceiverComSpec` belongs (for more details see [5]). This settings override possible settings in the `TransformationDescription` and `TransformationISignalProps`.

]([RS_SYST_00050](#))

[TPS_SYST_02075] Mandatory attributes in transformer configuration elements

[If a transformer configuration attribute is mandatory due to a particular constraint it means that it shall be defined in at least one of the three possible locations:

`TransformationDescription`, `TransformationISignalProps` or `TransformationComSpecProps`.]([RS_SYST_00050](#))

Please note that it is not required to define the complete attribute set on each of those locations. It means that it is allowed to overwrite single attributes in elements according to the precedence defined in [TPS_SYST_02074].

[TPS_SYST_02050] ISignal specific configuration of the SOME/IP Transformer

[The `ISignal` specific configuration of the SOME/IP Transformer shall be configured using `SOMEIPTransformationISignalProps`.]([RS_SYST_00050](#))

[TPS_SYST_02051] ISignal specific configuration of the E2E Transformer

[The `ISignal` specific configuration of the E2E Transformer shall be configured using `EndToEndTransformationISignalProps`.]([RS_SYST_00050](#))

For details how to configure those transformers `ISignal` specific please see chapter 7.3.2 and chapter 7.3.4.

[TPS_SYST_02052] ISignal specific configuration of custom transformers

[The `ISignal` specific configuration of custom transformers shall be configured using `UserDefinedTransformationISignalProps`.]([RS_SYST_00050](#))

To place the custom data in `UserDefinedTransformationDescription` the `AdminData` could be used for example.

To configure which communication shall be subject to transformation is done via references from `ISignals` and `ISignalGroups` to `DataTransformations`. An overview is shown in figure 7.10.

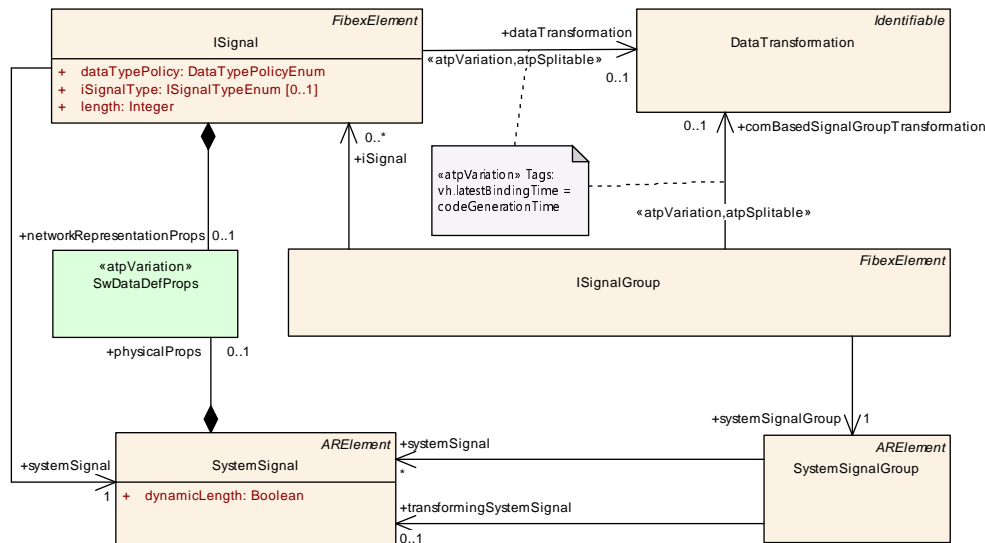


Figure 7.10: Configuration which communication shall be transformed

The `DataTransformation` element (which represents a transformer chain) is

- either referenced by the `ISignal` in the role `dataTransformation` which holds the transformed representation of the data
- referenced by the `ISignalGroup` in the role `comBasedSignalGroupTransformation` which holds the custom mapping of the data to the transformed representation or
- referenced by a `DataPrototypeMapping` in the role `firstToSecondDataTransformation`,

as defined in [constr_1400] in [5].

A `VariableDataPrototype` can either become a part of a `DataPrototypeMapping` based data transformation or of an `ISignal`-based data transformation as defined in [constr_1401] in [5].

[constr_1387] Transmission of Variable-Size Array Data Types by means of a Transformer [If a Transformer is used for the transmission of a Variable-Size Array Data Types then the Variable-Size Array Data Type shall be a “new-world” variable-size array data type according to [TPS_SWCT_01644] and [TPS_SWCT_01645]. “Old-world” dynamic-size array data types according to [TPS_SWCT_01642] and [TPS_SWCT_01643] are not supported.]()

7.3.1 Generic Transformer

[TPS_SYST_02053] A reference from `ISignal` to `DataTransformation` in the role `dataTransformation` enables data transformation [To enable the transformation of data, the `ISignal` which shall hold the transformed data shall reference a `DataTransformation` in the role `dataTransformation`.] (*RS_SYST_00050*)

[TPS_SYST_02054] Definition of data which shall be transformed [If

1. an `ISignal` references a `DataTransformation` and
2. this `ISignal` references a `SystemSignal` and
3. the referenced `SystemSignal` is referenced by a `SenderReceiverToSignalMapping` in the role `systemSignal` or referenced by a `ClientServerToSignalMapping` in the role `returnSignal` and in the role `callSignal`

then the `VariableDataPrototype` referenced by the `SenderReceiverToSignalMapping` or the `ClientServerOperation` referenced by the `ClientServerToSignalMapping` shall be transformed.] (*RS_SYST_00050*)

Using this configuration the result of the transformation will be put into the `ISignal` even if the data type is a composite type.

Furthermore, another `SystemSignal` can be added to a `SystemSignalGroup` in the role `transformingSystemSignal` to support the configuration where a complex data element is transferred via Sender/Receiver communication both using transformation and traditional mapping of RTE and COM.

The `ISignal` which references the `SystemSignal` which is referenced by a `SystemSignalGroup` in the role `transformingSystemSignal` shall reference a `DataTransformation` to transport the transformed data.

In parallel, the traditional mapping of RTE and COM maps all other `SystemSignals` of the `SystemSignalGroup` which are referenced in the role `systemSignal`.

[constr_3127] Certain `ISignals` always need a reference to `DataTransformation` [An `ISignal` which references a `SystemSignal` which is referenced by a `SystemSignalGroup` in the role `transformingSystemSignal` shall always reference a `DataTransformation`.] ()

7.3.2 SOME/IP Transformer

The specific configuration for SOME/IP transformers takes place in `SOMEIPTransformationDescription` and `SOMEIPTransformationISignalProps` shown in Figure 7.11.

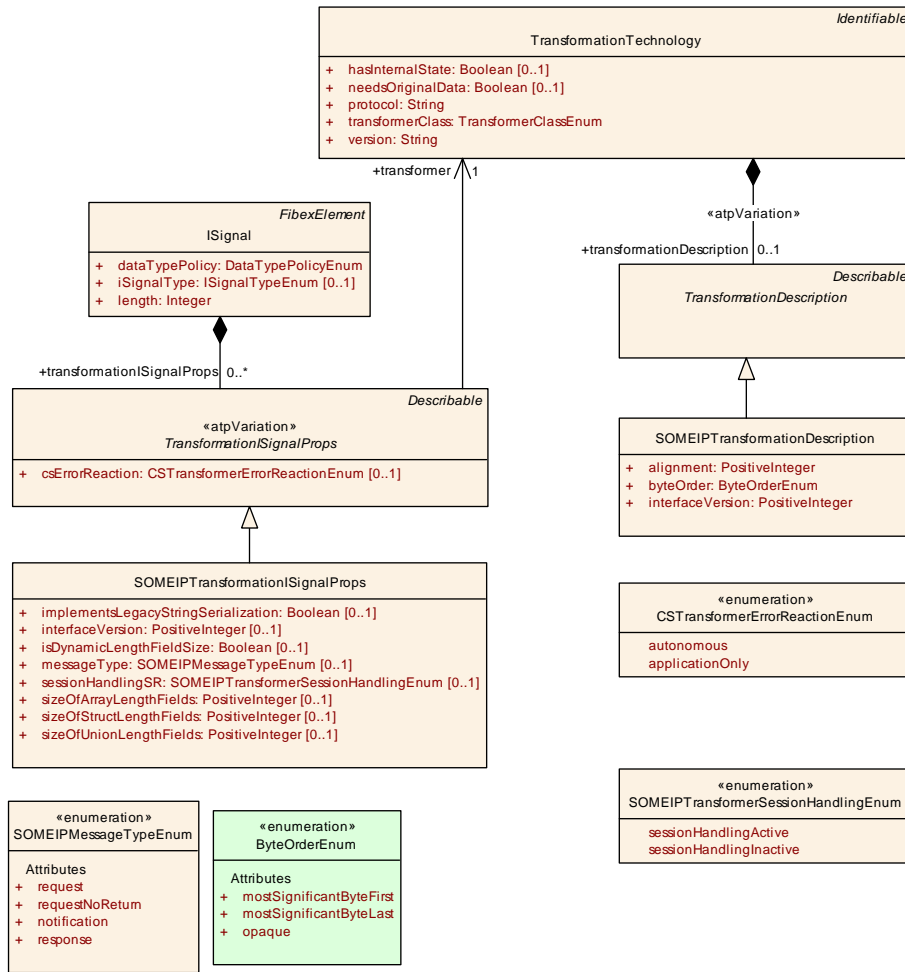


Figure 7.11: SOME/IP specific configuration

Class	SOMEIPTransformationDescription			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The SOMEIPTransformationDescription is used to specify SOME/IP transformer specific attributes.			
Base	ARObject, Describable, TransformationDescription			
Attribute	Type	Mult.	Kind	Note
alignment	PositiveInteger	1	attr	Specifies the alignment of dynamic data in the serialized data stream. The alignment shall be specified in Bits.
byteOrder	ByteOrderEnum	1	attr	Defines which byte order shall be serialized by the SOME/IP transformer
interfaceVersion	PositiveInteger	1	attr	The interface version the SOME/IP transformer shall use.

Table 7.10: SOMEIPTransformationDescription

Class	<<atpVariation>> SOMEIPTransformationISignalProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class SOMEIPTransformationISignalProps specifies ISignal specific configuration properties for SOME/IP transformer attributes.			
Base	<i>ARObject</i> , <i>Describable</i> , <i>TransformationISignalProps</i>			
Attribute	Type	Mult.	Kind	Note
implements LegacyString Serialization	Boolean	0..1	attr	This attribute indicates that Strings in the SOME/IP message shall NOT be serialized according to the SOME/IP specification for Strings. If this attribute is set to true, BOM and null-termination shall NOT be added in the serialization for Strings in the payload. If this attribute is set to false (or not set) BOM and null-termination shall be added in the serialization for Strings in the payload according to the SOME/IP specification for Strings. NOTE! This attribute is not future safe, and will be removed in an upcoming AUTOSAR release!"
interfaceVersion	PositiveInteger	0..1	attr	The interface version the SOME/IP transformer shall use.
isDynamic LengthFieldSize	Boolean	0..1	attr	This attribute shall be used to determine the wire type in the context of using the TLV encoding.
messageType	SOMEIPMessageType Enum	0..1	attr	The Message Type which shall be placed into the SOME/IP header.
session HandlingSR	SOMEIPTransformer SessionHandlingEnum	0..1	attr	Defines whether the SOME/IP transformer shall use session handling for Sender/Receiver communication.
sizeOfArray LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of fixed-size arrays in the SOME/IP message. This attribute is valid for all available occurrences of fixed-size arrays in the SOME/IP message. For a more fine granular modeling on the level of DataPrototypes the DataPrototypeTransformationProps shall be used.
sizeOfStruct LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of structs in the SOME/IP message. This attribute is valid for all available occurrences of structures in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
sizeOfUnion LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of unions in the SOME/IP message. This attribute is valid for all available occurrences of Unions in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
tlvDataId	TlvDataIdDefinitionSet	*	ref	This reference identifies the TlvDataIdDefinitions relevant for the enclosing SOMEIPTransformationISignalProps. Tags: atp.Status=draft

Table 7.11: SOMEIPTransformationISignalProps

Enumeration	ByteOrderEnum
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	When more than one byte is stored in the memory the order of those bytes may differ depending on the architecture of the processing unit. If the least significant byte is stored at the lowest address, this architecture is called little endian and otherwise it is called big endian. ByteOrder is very important in case of communication between different PUs or ECUs.
Literal	Description
mostSignificantByteFirst	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format) Tags: atp.EnumerationLiteralIndex=0
mostSignificantByteLast	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format) Tags: atp.EnumerationLiteralIndex=1
opaque	For opaque data endianness conversion has to be configured to Opaque. See AUTOSAR COM Specification for more details. Tags: atp.EnumerationLiteralIndex=2

Table 7.12: ByteOrderEnum

Enumeration	SOMEIPMessageTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Depending on the style of the communication different message types shall be set in the header of a SOME/IP message.
Literal	Description
notification	A request of a notification expecting no response. Tags: atp.EnumerationLiteralIndex=1
request	A request expecting a response. Tags: atp.EnumerationLiteralIndex=2
requestNoReturn	A fire&forget request. Tags: atp.EnumerationLiteralIndex=3
response	The response message. Tags: atp.EnumerationLiteralIndex=4

Table 7.13: SOMEIPMessageTypeEnum

Enumeration	SOMEIPTransformerSessionHandlingEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Enables or disable session handling for SOME/IP transformer
Literal	Description
sessionHandlingActive	The SOME/IP Transformer shall use session handling Tags: atp.EnumerationLiteralIndex=0
sessionHandlingInactive	The SOME/IP Transformer doesn't use session handling Tags: atp.EnumerationLiteralIndex=1

Table 7.14: SOMEIPTransformerSessionHandlingEnum

[constr_3128] SOME/IP transformer configuration [For each [TransformationDescription](#) variant that is a [SOMEIPTransformationDescription](#)

- attribute [protocol](#) of [TransformationTechnology](#) shall be set to SOMEIP

- attribute `version` of `TransformationTechnology` shall be set to `1.0.0`
- attribute `transformerClass` of `TransformationTechnology` shall be set to `serializer`
- attribute `headerLength` of `BufferProperties` shall be set to `64` (bits).

]()

The `SOMEIPTransformationDescription` contains the configuration for the transformer which shall be applied to all transformations. `ISignal` specific transformer configuration (which "override" the general ones) shall be done in `SOMEIPTransformationISignalProps`.

[TPS_SYST_02055] Alignment of SOME/IP [The attribute `alignment` defines the alignment used in the SOME/IP transformer in Bits.] ([RS_SYST_00050](#))

[TPS_SYST_02056] Byte Order of SOME/IP [The attribute `byteOrder` defines the byte order used in the SOME/IP transformer for creating the on wire format.] ([RS_SYST_00050](#))

[constr_3129] Byte Order of SOME/IP transformer [The attribute `byteOrder` of `SOMEIPTransformationDescription` shall be different from `opaque`.] ()

[TPS_SYST_02057] Interface Version of SOME/IP [The attribute `interfaceVersion` of `SOMEIPTransformationDescription` as well as `interfaceVersion` of `SOMEIPTransformationISignalProps` defines the interface version used by the SOME/IP transformer.] ([RS_SYST_00050](#))

[constr_3130] Range of Interface Version [The value of the attribute `interfaceVersion` shall be in the range `[0; 255]`] ()

[TPS_SYST_02092] Size of Fixed-size Array Length Fields [The attribute `sizeOfArrayLengthFields` of `SOMEIPTransformationISignalProps` defines the size of an length field generated by the SOME/IP transformer in front of all available fixed-size arrays in the `ISignal`. See also [\[constr_3282\]](#).] ([RS_SYST_00050](#))

[TPS_SYST_02093] Size of Structure Length Fields [The attribute `sizeOfStructLengthFields` of `SOMEIPTransformationISignalProps` defines the size of an length field generated by the SOME/IP transformer in front of all available structures in the `ISignal`. See also [\[constr_3283\]](#).] ([RS_SYST_00050](#))

[constr_1441] In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer [If an `ImplementationDataType` according to [\[TPS_SWCT_01700\]](#), i.e. of `category` `STRUCT` that encloses an `ImplementationDataTypeElement` of `category` `UNION`, is used to directly or (via a `DataTypeMap`) indirectly type an `AutosarDataPrototype` and the latter is mapped to a `SystemSignal` then the `ISignal` that references that `SystemSignal` shall aggregate `SOMEIPTransformationISignalProps` in the role `transformationISignalProps`.] ()

[TPS_SYST_02094] Size of Union Length Fields [The attribute `sizeOfUnionLengthFields` of `SOMEIPTransformationISignalProps` defines the size of a length field generated by the SOME/IP transformer in front of all available unions in the `ISignal`. See also [\[constr_3284\]](#).] (*RS_SYST_00050*)

In principle there is no need to define a size of the length indicator because the size can be computed from the data structure itself. However there is a use case to extend on the sender side while keeping the receiver side as it is. This means that there is the need to express the size of the length indicator because the extended data structure may reach a length that exceeds the capacity of the original computed size indicator.

[constr_3218] Range of Size of Fixed-size Array Length Fields [The value of attribute `sizeOfArrayLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.] ()

[constr_3220] Range of Size of Structure Length Fields [The value of attribute `sizeOfStructLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.] ()

[constr_3221] Range of Size of Union Length Fields [The value of attribute `sizeOfUnionLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.] ()

[TPS_SYST_02080] Message type of SOME/IP [The attribute `messageType` of `SOMEIPTransformationISignalProps` defines the message type used by the SOME/IP transformer for the serialized `ISignal`.] (*RS_SYST_00050*)

[constr_3216] Usage of `SOMEIPTransformationISignalProps.sessionHandlingSR` [The attribute `sessionHandlingSR` of `SOMEIPTransformationISignalProps` shall only be used for `ISignals` which reference `SystemSignals` which are mapped via a `SenderReceiverToSignalMapping`.] ()

Note: This means that `sessionHandlingSR` shall only be used for transformed Sender/Receiver communication.

7.3.2.1 Handling of Strings

Prior to AUTOSAR Release 4.3 the SOME/IP Transformer did not specify any special handling for how String data types shall be serialized. This leads to the situation that no BOM is introduced. The SOME/IP standard however does expect a BOM for String data type, this is also how AUTOSAR Release 4.3 defines the serialization of Strings.

The attribute `SOMEIPTransformationISignalProps.implementsLegacyStringSerialization` enables the compatibility of string handling between AUTOSAR pre Release 4.3 and standard SOME/IP serializers. An AUTOSAR ECU according to Release 4.2 would not define any String specific behavior, thus it is possible to enforce this for later Release implementations as well by setting the attribute to true.

7.3.2.2 SOME/IP Transformation Properties on the level of DataPrototypes

The serialization of SOME/IP is based on the interface specification. For certain datatypes like structures, unions and arrays SOME/IP supports the configuration of length fields that will be put in front of the serialized data. AUTOSAR supports the configuration of such SOME/IP settings on two different levels:

- modeling on `ISignal` level that is valid for all available occurrences of a datatype in the SOME/IP message (see [TPS_SYST_02092], [TPS_SYST_02093] and [TPS_SYST_02094])
- fine granular modeling on the level of `DataPrototypes` (see [TPS_SYST_02121])

To allow such a fine granular modeling `SOMEIPTransformationProps` are defined and collected in `TransformationPropsSets`.

Class	TransformationPropsSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Collection of TransformationProps. Tags: atp.recommendedPackage=TransformationPropsSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
transformation Props	TransformationProps	*	aggr	Transformer specific configuration properties.

Table 7.15: TransformationPropsSet

Class	TransformationProps (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This meta-class represents a abstract base class for transformation settings.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	SOMEIPTransformationProps , UserDefinedTransformationProps			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 7.16: TransformationProps

Class	SOMEIPTransformationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class SOMEIPTransformationProps specifies SOME/IP specific configuration properties.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable , TransformationProps			
Attribute	Type	Mult.	Kind	Note
alignment	PositiveInteger	0..1	attr	Specifies the alignment of dynamic data in the serialized data stream. The alignment is specified in Bits.





Class	SOMEIPTransformationProps			
sizeOfArrayLengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a static size Array in the SOME/IP message.
sizeOfStructLengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Structure in the SOME/IP message.
sizeOfUnionLengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Union in the SOME/IP message.

Table 7.17: SOMEIPTransformationProps

The relation between `SOMEIPTransformationProps` and a `DataPrototype` is created with `DataPrototypeTransformationProps` in the context of an `ISignal`.

[TPS_SYST_02127] Usage of `DataPrototypeTransformationProps` in case of a `VariableDataPrototype` [If a `VariableDataPrototype` is transported in the `ISignal` the `DataPrototypeTransformationProps` can be used to assign `SOMEIPTransformationProps` to a `DataPrototype` that is or is part of the `VariableDataPrototype`.] (*RS_SYST_00050*)

[TPS_SYST_02128] Usage of `DataPrototypeTransformationProps` in case of a `ClientServerOperation` [If a `ClientServerOperation` is transported in the `ISignal` (`callSignal` or `returnSignal`) the `DataPrototypeTransformationProps` can be used to assign `SOMEIPTransformationProps` to a `DataPrototype` that is or is part of an `ArgumentDataPrototype` of the `ClientServerOperation`.] (*RS_SYST_00050*)

[TPS_SYST_02129] Assignment of `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `SenderReceiverInterface` typed by an `ApplicationDataType` [To assign the `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `SenderReceiverInterface` that is typed by an `ApplicationDataType` the `DataPrototypeInSenderReceiverInterfaceInstanceRef` shall reference the `DataPrototype` with the `targetDataPrototypeInSr` reference. The `rootDataPrototypeInSr` and `contextDataPrototypeInSr` references shall not be used.] (*RS_SYST_00050*)

[TPS_SYST_02212] Assignment of `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `ClientServerInterface` typed by an `ApplicationDataType` [To assign the `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `ClientServerInterface` that is typed by an `ApplicationDataType` the `DataPrototypeInClientServerInterfaceInstanceRef` shall reference the `DataPrototype` with the `targetDataPrototypeInCs` reference. The `rootDataPrototypeInCs` and `contextDataPrototypeInCs` references shall not be used.] (*RS_SYST_00050*)

[TPS_SYST_02130] Assignment of `SOMEIPTransformationProps` to a subElement of a root `AutosarDataPrototype` in a `SenderReceiverInterface` typed by an `ApplicationDataType` [To assign the `SOMEIPTransformationProps` to

a subElement of a root `AutosarDataPrototype` in a `SenderReceiverInterface` that is typed by an `ApplicationDataType` the `DataPrototypeInSenderReceiverInterfaceInstanceRef` shall reference the subElement with the `targetDataPrototypeInSr` reference. In addition the `rootDataPrototypeInSr` shall be set to define the context. Optionally it may be necessary to use `contextDataPrototypeInSr` references because the target subElement may be arbitrarily nested within the root `AutosarDataPrototype`.] (*RS_SYST_00050*)

[TPS_SYST_02213] Assignment of `SOMEIPTransformationProps` to a subElement of a root `AutosarDataPrototype` in a `ClientServerInterface` typed by an `ApplicationDataType` [To assign the `SOMEIPTransformationProps` to a subElement of a root `AutosarDataPrototype` in a `ClientServerInterface` that is typed by an `ApplicationDataType` the `DataPrototypeInClientServerInterfaceInstanceRef` shall reference the subElement with the `targetDataPrototypeInCs` reference. In addition the `rootDataPrototypeInCs` shall be set to define the context. Optionally it may be necessary to use `contextDataPrototypeInCs` references because the target subElement may be arbitrarily nested within the root `AutosarDataPrototype`.] (*RS_SYST_00050*)

[TPS_SYST_02131] Assignment of `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `SenderReceiverInterface` typed by an `ImplementationDataType` [To assign the `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `SenderReceiverInterface` that is typed by an `ImplementationDataType` the `DataPrototypeInSenderReceiverInterfaceInstanceRef` shall reference the `AutosarDataPrototype` with the `targetDataPrototypeInSr` reference. The `rootDataPrototypeInSr` and `contextDataPrototypeInSr` references in the `DataPrototypeInSenderReceiverInterfaceInstanceRef` shall not be used.] (*RS_SYST_00050*)

[TPS_SYST_02214] Assignment of `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `ClientServerInterface` typed by an `ImplementationDataType` [To assign the `SOMEIPTransformationProps` to a root `AutosarDataPrototype` in a `ClientServerInterface` that is typed by an `ImplementationDataType` the `DataPrototypeInClientServerInterfaceInstanceRef` shall reference the `AutosarDataPrototype` with the `targetDataPrototypeInCs` reference. The `rootDataPrototypeInCs` and `contextDataPrototypeInCs` references in the `DataPrototypeInClientServerInterfaceInstanceRef` shall not be used.] (*RS_SYST_00050*)

[TPS_SYST_02132] Assignment of `SOMEIPTransformationProps` to a subElement of a root `AutosarDataPrototype` typed by an `ImplementationDataType`

[To assign the `SOMEIPTransformationProps` to a subElement of a root `AutosarDataPrototype` in a `ClientServerInterface` or `SenderReceiverInterface` that is typed by an `ImplementationDataType` the `ImplementationDataTypeElementInPortInterfaceRef` shall reference the `targetImplementationDataTypeElement`. In addition the `rootDataPrototype` shall be set to define the context. Optionally it may be necessary to use `contextImplementationDataElement` references because the target subElement may be arbitrarily nested within the root `AutosarDataPrototype`.] (*RS_SYST_00050*)

[**TPS_SYST_02195**] **Applicable use cases for `DataPrototypeReference`** [Table 7.18 contains a comprehensive list of use cases for the usage of `DataPrototypeReference`.] (*RS_SYST_00050*)

Use case	Role
<code>AutosarDataPrototype</code> in a <code>SenderReceiverInterface</code> typed by an <code>ApplicationDataType</code>	<code>DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr</code>
<code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ApplicationDataType</code>	<code>DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs</code>
<code>DataPrototype</code> in <code>AutosarDataPrototype</code> in a <code>SenderReceiverInterface</code> typed by an <code>ApplicationCompositeDataType</code>	<code>DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr</code>
<code>DataPrototype</code> in <code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ApplicationCompositeDataType</code>	<code>DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs</code>
<code>AutosarDataPrototype</code> in a <code>SenderReceiverInterface</code> typed by an <code>ImplementationDataType</code>	<code>DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr</code>
<code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ImplementationDataType</code>	<code>DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs</code>
<code>DataPrototype</code> in <code>AutosarDataPrototype</code> typed by an <code>ImplementationDataType</code>	<code>ImplementationDataTypeElementInPortInterfaceRef.targetImplementationDataTypeElement</code>

Table 7.18: Possible use cases for the usage of `DataPrototypeReference`

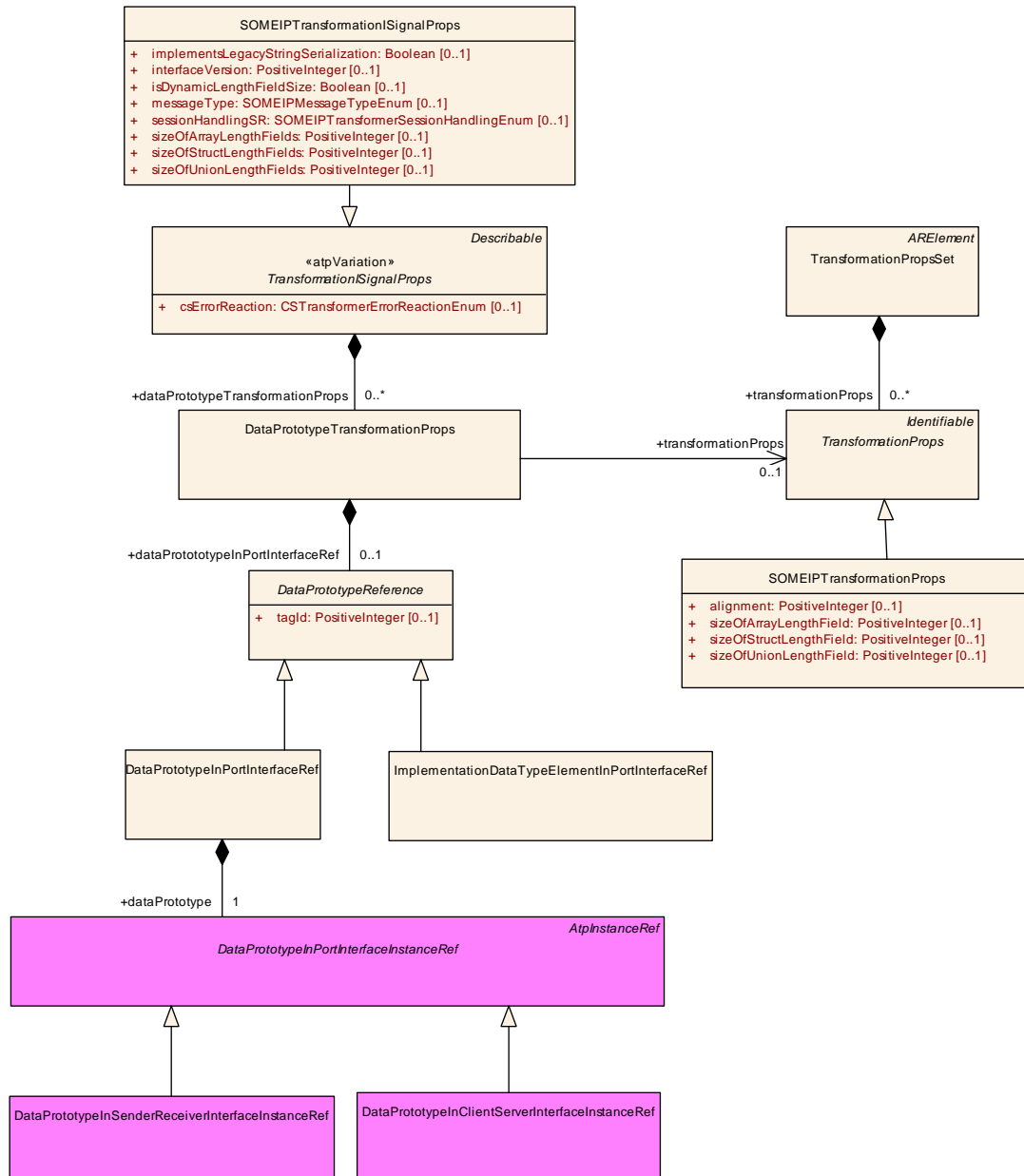


Figure 7.12: Transformation Properties on the level of DataPrototypes

Class	DataPrototypeTransformationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	DataPrototypeTransformationProps allows to set the attributes for the different Transformation Technologies that are DataPrototype specific.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
dataPrototypeInPortInterfaceRef	DataPrototypeReference	0..1	aggr	Reference to a DataPrototype that is transported in the serialized ISignal.





Class	DataPrototypeTransformationProps			
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation for the referenced primitive DataPrototype. If a network representation is provided then the baseType shall be used by the Transformer as input for the serialization/deserilaization.
transformation Props	TransformationProps	0..1	ref	Collection of AutosarDataPrototype related configuration settings for a transformer.

Table 7.19: DataPrototypeTransformationProps

Class	DataPrototypeReference (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This meta-class provides the ability to reference a DataPrototype.			
Base	ARObject			
Subclasses	DataPrototypeInPortInterfaceRef , ImplementationDataTypeElementInPortInterfaceRef			
Attribute	Type	Mult.	Kind	Note
tagId	PositiveInteger	0..1	attr	This attribute represents the ability to specify a tag-id for the serialization of a specific DataPrototype in the context of a (potentially deeply-nested) composite data structure.

Table 7.20: DataPrototypeReference

Class	DataPrototypeInPortInterfaceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This class represents a RootDataPrototype that is typed by an ApplicationDataType or ImplementationDataType or a DataTypeElement that is aggregated within a composite application data type (record or array).			
Base	ARObject, DataPrototypeReference			
Attribute	Type	Mult.	Kind	Note
dataPrototype	DataPrototype	1	iref	This element defines a reference to a DataPrototype in the context of a PortInterface.

Table 7.21: DataPrototypeInPortInterfaceRef

Class	DataPrototypeInSenderReceiverInterfaceInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note	Tags:atp.Status=draft			
Base	ARObject, AtpInstanceRef , DataPrototypeInPortInterfaceInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	SenderReceiverInterface	0..1	ref	Stereotypes: atpDerived
contextData PrototypelnSr (ordered)	ApplicationCompositeElementDataPrototype	*	ref	Tags: xml.sequenceOffset=20
rootData PrototypelnSr	AutosarDataPrototype	0..1	ref	Tags: xml.sequenceOffset=10





Class	DataPrototypeInSenderReceiverInterfaceInstanceRef			
targetData PrototypeInSr	DataPrototype	1	ref	Tags:xml.sequenceOffset=30

Table 7.22: DataPrototypeInSenderReceiverInterfaceInstanceRef

Class	DataPrototypeInClientServerInterfaceInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note	Tags:atp.Status=draft			
Base	<i>ARObject</i> , AtpInstanceRef , DataPrototypeInPortInterfaceInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	ClientServerInterface	0..1	ref	Stereotypes: atpDerived
contextData PrototypeInCs	ApplicationComposite ElementDataPrototype	*	ref	Tags:xml.sequenceOffset=20
rootData PrototypeInCs	AutosarDataPrototype	0..1	ref	Tags:xml.sequenceOffset=10
targetData PrototypeInCs	DataPrototype	1	ref	Tags:xml.sequenceOffset=30

Table 7.23: DataPrototypeInClientServerInterfaceInstanceRef

Class	ImplementationDataTypeElementInPortInterfaceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note	This meta-class represents the ability to refer to the internal structure of an AutosarDataPrototype which is typed by an ImplementationDatatype in the context of a PortInterface. In other words, this meta-class shall not be used to model a reference to the AutosarDataPrototype as a target itself, even if the AutosarDataPrototype is typed by an ImplementationDataType and even if that ImplementationDataType represents a composite data type.			
Base	<i>ARObject</i> , DataPrototypeReference			
Attribute	Type	Mult.	Kind	Note
context Implementation DataElement (ordered)	ImplementationData TypeElement	*	ref	This is a context in case there are subelements with explicit types. The reference has to be ordered to properly reflect the nested structure. Tags:xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	This refers to the AutosarDataPrototype which is typed by the ImplementationDatatype. The targetDataPrototype and all defined contextDataPrototypes can be found within this rootDataPrototype. Tags:xml.sequenceOffset=10
target Implementation DataType Element	ImplementationData TypeElement	0..1	ref	This is a target ImplementationDataTypeElement in case that the rootDataPrototype is composite and the target is a subElement of the rootDataPrototype. Tags:xml.sequenceOffset=30

Table 7.24: ImplementationDataTypeElementInPortInterfaceRef

[TPS_SYST_02121] Scope of [DataPrototypeTransformationProps](#) [[DataPrototypeTransformationProps](#)] is defined either

- for the root [DataPrototype](#) that is transmitted in the serialized [ISignal](#)
- for each of the composite subElements of the composite root [DataPrototype](#)

](RS_SYST_00050)

[TPS_SYST_02123] Size of a length field for a chosen fixed-size array [The attribute `sizeOfArrayLengthField` of `SOMEIPTransformationProps` defines the size of a length field generated by the SOME/IP transformer in front of the fixed-size array for which the `DataPrototypeTransformationProps` is defined according to [TPS_SYST_02121].](RS_SYST_00050)

[TPS_SYST_02124] Size of a length field for a chosen structure [The attribute `sizeOfStructLengthField` of `SOMEIPTransformationProps` defines the size of a length field generated by the SOME/IP transformer in front of the structure for which the `DataPrototypeTransformationProps` is defined according to [TPS_SYST_02121].](RS_SYST_00050)

[TPS_SYST_02125] Size of a length field for a chosen union [The attribute `sizeOfUnionLengthField` of `SOMEIPTransformationProps` defines the size of a length field generated by the SOME/IP transformer in front of the union for which the `DataPrototypeTransformationProps` is defined according to [TPS_SYST_02121].](RS_SYST_00050)

[TPS_SYST_02126] Alignment of a dynamic DataPrototype [The attribute `alignment` of `SOMEIPTransformationProps` defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element for which the `DataPrototypeTransformationProps` is defined according to [TPS_SYST_02121].](RS_SYST_00050)

[constr_3278] Usage of `SOMEIPTransformationProps.sizeOfArrayLengthField` [The attribute `sizeOfArrayLengthField` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a static size array according to [TPS_SYST_02121].]()

[constr_3279] Usage of `SOMEIPTransformationProps.sizeOfStructLengthField` [The attribute `sizeOfStructLengthField` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a structure according to [TPS_SYST_02121].]()

[constr_3280] Usage of `SOMEIPTransformationProps.sizeOfUnionLengthField` [The attribute `sizeOfUnionLengthField` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a union according to [TPS_SYST_02121].]()

[constr_3281] Usage of `SOMEIPTransformationProps.alignment` [The attribute `alignment` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a variable data length data element according to [TPS_SYST_02121].]()

[constr_3282] SOME/IP Transformation settings for static size arrays in the context of an `ISignal` [In the context of an `ISignal` the usage

of `DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfArrayLengthFields` is not defined.]()

[constr_3283] SOME/IP Transformation settings for structures in the context of an ISignal [In the context of an `ISignal` the usage of `DataPrototypeTransformationProps.transformationProps.sizeOfStructLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfStructLengthFields` is not defined.]()

[constr_3284] SOME/IP Transformation settings for unions in the context of an ISignal [In the context of an `ISignal` the usage of `DataPrototypeTransformationProps.transformationProps.sizeOfUnionLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfUnionLengthFields` is not defined.]()

[constr_3285] Alignment of variable data length data elements in the context of an ISignal [The definition of `DataPrototypeTransformationProps.transformationProps.alignment` is only allowed if the `SOMEIPTransformationDescription.alignment` is not defined.]()

7.3.2.3 Network Representation

In order to assure that the serialization of the transported data on the sender side and its deserialization on the receiver side(s) is done correctly, system designers need to assure that the same datatypes (i.e., `SwBaseTypes`) are used for the serialization/deserialization on both sides. However, this agreement does not imply the use or equality of the `SwBaseTypes` defined by the `ImplementationDataType` used by the application software on the sender and (possibly multiple) receiver sides. This means that each `EcuInstance`, regardless if it belongs to a sender or receiver, can use one datatype for the serialization/deserialization (e.g., `UInt16` in the actual SOME/IP transformer code) and another datatype in the application software (e.g., `Float32` in the actual application software component code).

In order to define the commonly agreed datatypes for the serialization/deserialization of the transported data by the sender and possibly multiple receivers, AUTOSAR defines the following two approaches:

- serialization based on the network representation ([TPS_SYST_02136])
- serialization based on the `ImplementationDataTypes` ([TPS_SYST_02137])

[TPS_SYST_02136] Serialization based on the network representation [If a network representation that defines a `SwBaseType` and optionally a `CompuMethod` is provided for each `DataPrototype` typed by a primitive data type that is part of the serialized `ISignal` (`ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps`), these `SwBaseTypes` shall be used for the serialization/deserialization.]()

[TPS_SYST_02137] Serialization based on the [ImplementationDataTypes](#) [For primitive [DataPrototype](#)s that are part of the serialized [ISignal](#) where no network representation is provided ([ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps](#)), [SwBaseType](#) shall be provided by the [ImplementationDataTypes](#) that either types the corresponding [PortPrototypes](#) on the top level Software Composition that represents the communicating [EcuInstances](#), or it is mapped to the [ApplicationDataType](#) that types it.]()

[constr_3317] Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the [ImplementationDataTypes](#) [In order to assure the same interpretation of the serialized data by the SOME/IP transformers on the sender and receiver sides in case of serialization based on either a primitive or a composite [ImplementationDataType](#), the same [SwBaseType](#) shall be defined

- for this primitive [DataPrototype](#) or
- for each primitive [DataPrototype](#) of the leaf elements of the composite [DataPrototype](#) starting from the first element until and including the last element that is requested by the receiver,

by the [ImplementationDataTypes](#) that either types the corresponding [PortPrototypes](#) on the top level Software Composition of the communicating [EcuInstances](#), or it is mapped to the [ApplicationDataType](#) that types it.]()

If the serialization is based on the [ImplementationDataTypes](#), the same data has to be transmitted on all buses, i.e., it is not possible to transmit different precision (i.e., number of bits) on different buses, as with the serialization based on the network representation on the [ISignal](#) level.

[ImplementationDataTypes](#) used by the actual application for the transported data shall be defined by the corresponding [PortPrototypes](#) on the [AtomicSwComponentTypes](#) of the communicating [EcuInstances](#). The RTE is responsible for the possible type conversion and scaling in case of different [ImplementationDataType](#)s used for the serialization/deserialization and in the application.

[TPS_SYST_02138] Definition of the network representation [The network representation for each [DataPrototype](#) typed by a primitive data type in the serialized data shall be defined by the [SwDataDefProps](#) that is aggregated by the [DataPrototypeTransformationProps](#) in the role [networkRepresentationProps](#).]()

In other words: If a [DataPrototype](#) is transported in the [ISignal](#) the [DataPrototypeTransformationProps](#) can be used to assign a network representation to each primitive [DataPrototype](#) that is part of the enclosing [DataPrototype](#).

Attributes of SwDataDefProps	networkRepresentationProps
additionalNativeTypeQualifier	NA
annotation	NA
baseType	D
compuMethod	D
dataConstr	D

displayFormat	D
displayPresentation	NA
implementationDataType	NA
invalidValue	NA
swAddrMethod	NA
swAlignment	NA
swBitRepresentation	NA
swCalibrationAccess	NA
swCalprmAxisSet	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. swCalprmRef	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. swVariableRef	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. sharedAxisType	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. inputVariableType	NA
swCalprmAxisSet/ AxisIndividual/ Unit	NA
swCalprmAxisSet/ BaseType	NA
swComparisonVariable	NA
swDataDependency	NA
swHostVariable	NA
swImplPolicy	NA
swIntendedResolution	NA
swInterpolationMethod	NA
swIsVirtual	NA
swPointerTargetProps	NA
swRecordLayout	NA
swRefreshTiming	NA
swTextProps	NA
swValueBlockSize	NA
unit	D
valueAxisDataType	NA

Table 7.25: Allowed SwDataDefProps Attributes on DataPrototypeTransformationProps

The following settings apply in table 7.25:

D Attribute can be **defined** in the scope of this element.

NA Attribute is **not applicable** for usage in the scope of this element.

[TPS_SYST_02139] Applicability of the SwDataDefProps attributes for the network representation of the serialized data [Usage of `DataPrototypeTransformationProps.networkRepresentationProps` shall follow the restrictions given in table Table 7.25.]()

[constr_3318] Allowed use of ISignal.networkRepresentationProps [If a reference from `ISignal` to `DataTransformation` in the role `dataTransformation` exists, this `ISignal` SHALL NOT aggregate `SwDataDefProps` in the role `networkRepresentationProps`.]()

This means that aggregating `SwDataDefProps` by an `ISignal` is applicable only if this `ISignal` is not transformed.

[constr_3319] Existence of `DataPrototypeTransformationProps.networkRepresentationProps` [`ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps`] shall either

- not exist at all or
- shall be defined for all leaf elements of the root `DataPrototype` transmitted in the `ISignal`

]()

This means that either all leaf elements of the transformed `ISignal` shall have a network representation, or none.

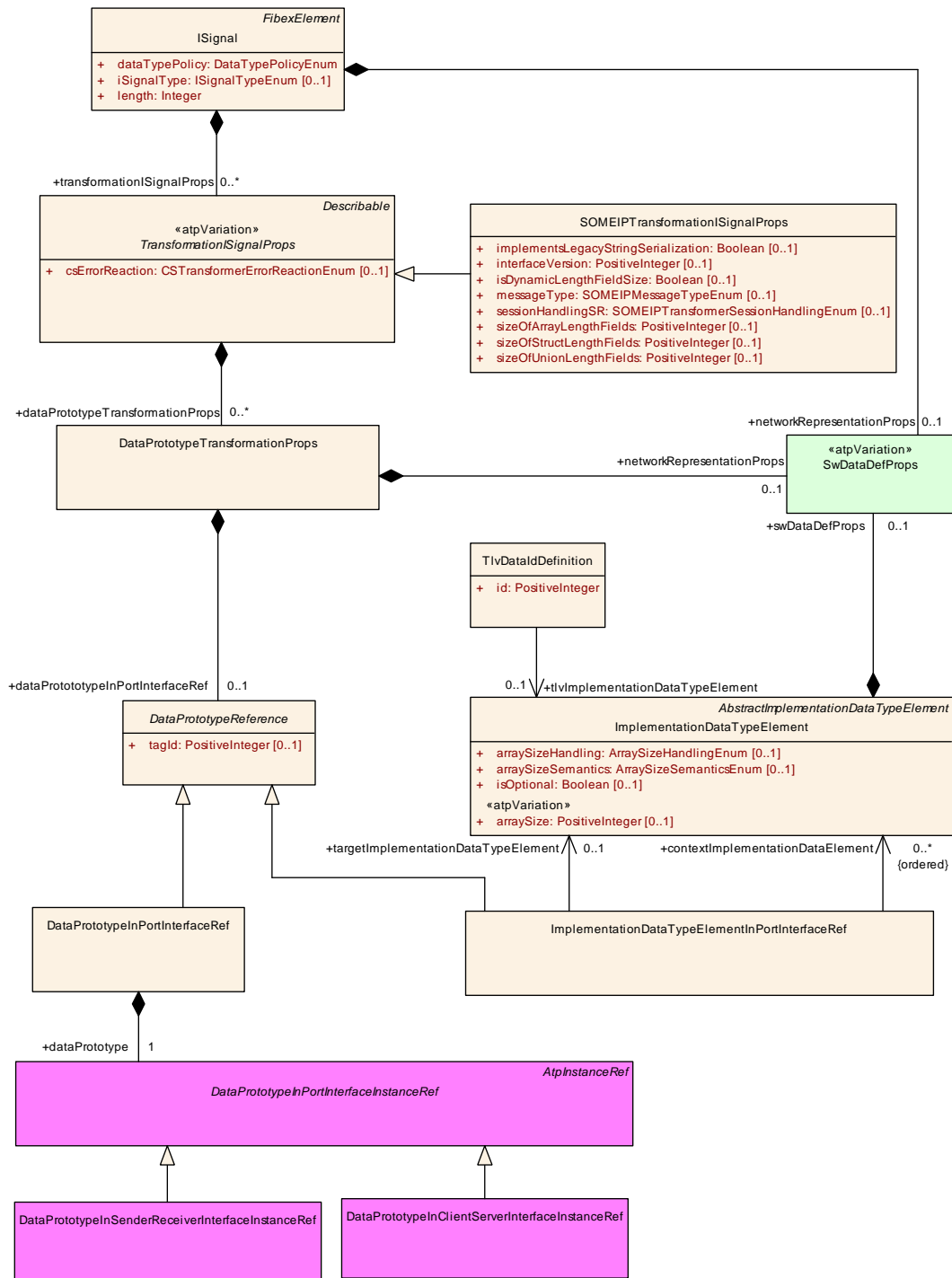


Figure 7.13: Transformer Network Representation

7.3.2.3.1 Example - Serialization based on the network representation

An example with concrete methodological steps in a common OEM-Tier 1 development process for the serialization based on the network representation is presented in [Figure 7.14](#). The steps are as follows:

1. OEM decides on a common `SwBaseType` and `CompuMethod` for each bus, as part of the network representation, used for serialization/deserialization of one concrete complex data type.
2. OEM provides an `ImplementationDataType`, with `SwBaseType` and optional `CompuMethod`, on the `PortPrototypes` on the `RootSwCompositions` of the communicating `EcuInstances` (sender and possibly multiple receivers). The step is optional and `PortPrototypes` can also be typed by an `ApplicationDataType` that has a mapping to an `ImplementationDataType`.
3. Tier 1s are free to define arbitrary `ImplementationDataType` (with `SwBaseType` and optional `CompuMethod`) in the application software components. If this `SwBaseType` is different than the one used for the serialization/deserialization, RTE is responsible for the type conversion together with possible scaling defined by the `CompuMethods`, as part of the network representation and `PortPrototypes` on the `RootSwComposition` and `SwComponentPrototype` that is typed by `ApplicationSwComponentType`. Please note that on the receiver side it is possible that the `SwComponentPrototype` that is typed by `ApplicationSwComponentType` receives only a subset of data defined on the `RootSwComposition`. In this case, this needs to be described by the `PortInterfaceMapping`.

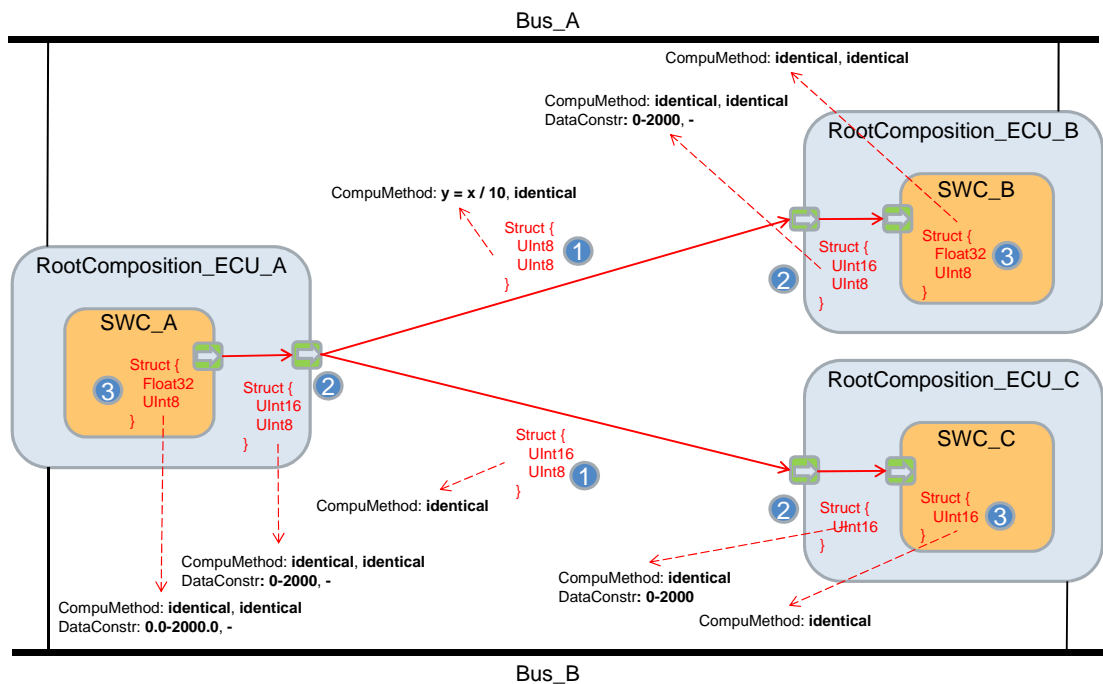


Figure 7.14: Serialization based on the network representation

The actual steps that need to be performed at runtime are presented in [Figure 7.15](#) and they are as follows:

1. Application of the sending software component provides the data to be transmitted to the RTE and stores it in SWC internal buffer.

2. If `SwBaseType` defined by the `ImplementationDataType` on the `PortPrototype` of the `ApplicationSwComponentType` is different then the one optionally defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`, the RTE performs type conversion, and scaling if `CompuMethods` are also defined on these `PortPrototypes`, and stores the values internally in the RTE.
3. If network representation defines a `SwBaseType` that is different from the one optionally defined by the `ImplementationDataType` on the `PortPrototype` on the `RootSwComposition`, the RTE performs another type conversion, and scaling if `CompuMethod` is also defined as part of the network representation, and stores the value internally in the RTE. This internal value is used for the serialization.
4. On the receiver side, the RTE stores the serialized data in the RTE internal buffer. When the receiver SWC wants to read the data, the RTE first de-serializes the values received from the bus whose type is specified by the `SwBaseType` that is part of the network representation. If the `SwBaseType` is different then the one optionally defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`, the RTE performs type conversion, and scaling if `CompuMethods` are also defined on the `PortPrototype` and in the network representation, and stores the values internally in the RTE.
5. If `SwBaseType` defined by the `ImplementationDataType` on the `PortPrototype` of the `SwComponentPrototype` that is typed by `ApplicationSwComponentType` is different then the one defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`, the RTE performs another type conversion, and scaling if `CompuMethods` are also defined on these `PortPrototype`, and stores the final values internally in the buffer of the application. The RTE is also able to deliver only a subset of data to the `SwComponentPrototype` that is typed by `ApplicationSwComponentType`, if that is required by the description of the `PortInterfaceMapping`.

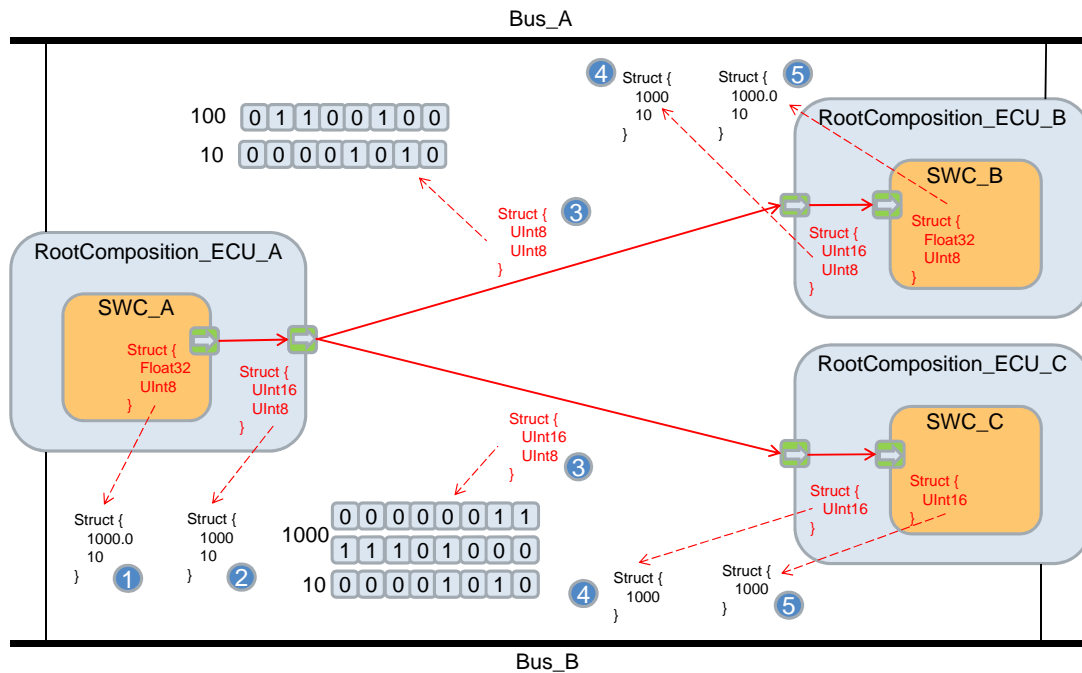


Figure 7.15: Serialization based on the network representation

7.3.2.3.2 Example - Serialization based on the ImplementationDataTypes

An example with concrete methodological steps in a common OEM-Tier 1 development process for the serialization based on the [ImplementationDataTypes](#) is presented in [Figure 7.16](#). The steps are as follows:

1. OEM provides the same [ImplementationDataType](#), with [SwBaseType](#) and optional [CompuMethod](#), on the [PortPrototypes](#) on the [RootSwCompositions](#) of the communicating [EcuInstances](#) (sender and possibly multiple receivers). The [PortPrototypes](#) can also be typed by an [ApplicationDataType](#) that has a mapping to an [ImplementationDataType](#).
2. Tier 1s are free to define arbitrary [ImplementationDataType](#) (with [SwBaseType](#) and optional [CompuMethod](#)) in the application software components. If this [SwBaseType](#) is different than the one used for the serialization/deserialization, RTE is responsible for the type conversion together with possible scaling defined by the [CompuMethods](#), as part of [PortPrototypes](#) on the [RootSwComposition](#) and [AtomicSwComponentTypes](#). Please note that on the receiver side it is possible that the [SwComponentPrototype](#) that is typed by [ApplicationSwComponentType](#) receives only a subset of data defined on the [RootSwComposition](#). In this case, this needs to be described by the [PortInterfaceMapping](#).

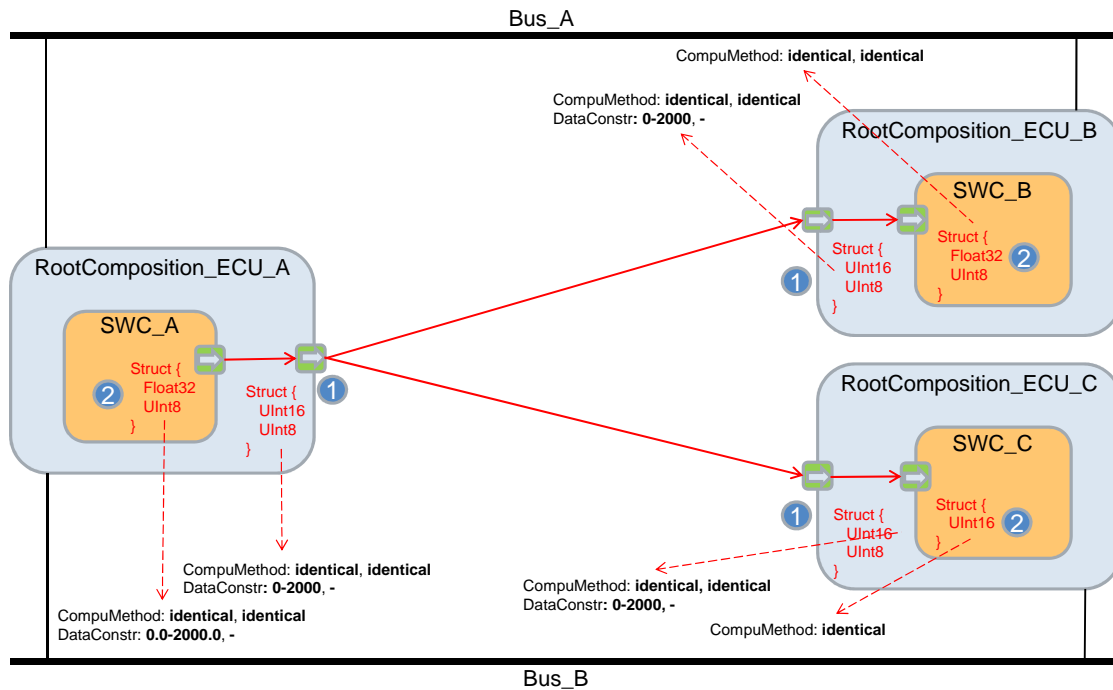


Figure 7.16: Serialization based on the ImplementationDataTypes

The actual steps that need to be performed at runtime are presented in [Figure 7.17](#) and they are as follows:

1. Application of the sending software component provides the data to be transmitted to the RTE and stores it in SWC internal buffer.
2. If `SwBaseType` defined by the `ImplementationDataType` on the `PortPrototype` of the `SwComponentPrototype` that is typed by `ApplicationSwComponentType` is different then the one defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`, the RTE performs type conversion, and scaling if `CompuMethods` are also defined on these `PortPrototype`s, and stores the values internally in the RTE.
3. As no network representation is provided, the internal value from step 2 is used for the serialization and transmission on the bus.
4. On the receiver side, the RTE stores the serialized data received from the bus in the RTE internal buffer. When the receiver SWC wants to read the data, the RTE de-serializes these values as defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`.
5. If `SwBaseType` defined by the `ImplementationDataType` on the `PortPrototype` of the `SwComponentPrototype` that is typed by `ApplicationSwComponentType` is different then the one defined by the `ImplementationDataType` on the `PortPrototype` of the `RootSwComposition`, the RTE performs another type conversion, and scaling if `CompuMethods` are also defined on these `PortPrototype`, and stores the final values internally in the buffer of the application. The RTE is also able to deliver only a subset of data to the

`SwComponentPrototype` that is typed by `ApplicationSwComponentType`, if that is required by the description of the `PortInterfaceMapping`.

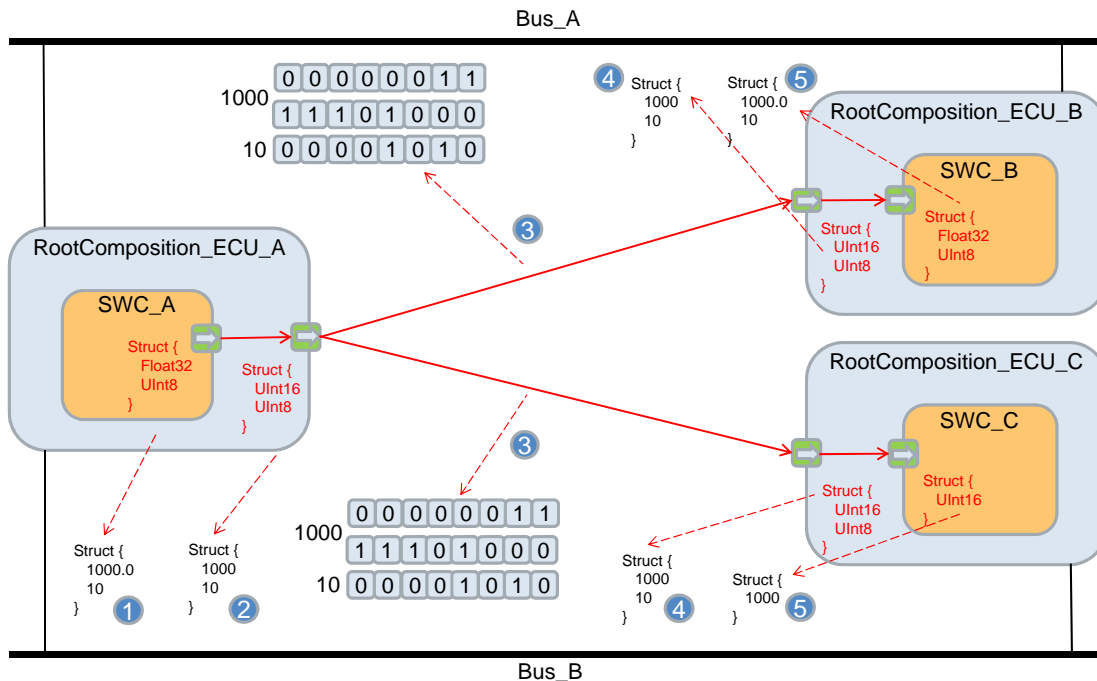


Figure 7.17: Serialization based on the ImplementationDataTypes

7.3.3 COM Based Transformer

In order to support the signal group based interaction of the transformer with the COM module as defined in the COM Based Transformer specification [17] a further modeling is supported:

In case the array based signal group API of Com shall be used the `ISignalGroup` has a reference to the `DataTransformation` element in the role `comBasedSignalGroupTransformation`. This defines that the RTE shall use the array based signal group API of Com [21] in order to transport the transformed data.

[TPS_SYST_02058] Usage of COM Based Transformer [If

1. an `ISignalGroup` references a `DataTransformation` in the role `comBasedSignalGroupTransformation` and
2. this `ISignalGroup` references a `SystemSignalGroup` and
3. the referenced `SystemSignalGroup` is referenced by a `SenderReceiverToSignalGroupMapping` in the role `signalGroup`

then the `VariableDataPrototype` referenced by the `SenderReceiverToSignalGroupMapping` shall be transformed using the COM Based Transformer [17].
(*RS_SYST_00051*)

[constr_5050] VariableDataPrototype of COM Based Transformer [The `VariableDataPrototype` of `[TPS_SYST_02058]` shall be typed by an `ApplicationRecordDataType` or an `ImplementationDataType` of category STRUCTURE.]
()

[TPS_SYST_02133] BufferProperties.bufferComputation setting for a COM Based transformer [The `BufferProperties.bufferComputation` of a COM Based transformer shall be configured in the following way:

```
<BUFFER-COMPUTATION>
  <COMPU-RATIONAL-COEFFS>
    <COMPU-NUMERATOR>
      <V>{Position of last byte in ISignalGroup - Position of first byte in
        ISignalGroup} + 1</V>
      <V>0</V>
    </COMPU-NUMERATOR>
    <COMPU-DENOMINATOR>
      <V>1</V>
    </COMPU-DENOMINATOR>
  </COMPU-RATIONAL-COEFFS>
</BUFFER-COMPUTATION>
```

]()

Please note that according to `[SWS_Rte_03867]` the RTE calculates the `InputBufferLength` that is used for the output buffer calculation for Sender/Receiver communication needed for the `VariableDataPrototype` of the `dataElement` of the `SenderReceiverInterface` that shall be transformed.

[constr_3315] The value of V0 in BufferProperties.bufferComputation setting for a COM Based transformer [The value of V0 of `bufferComputation` of a `TransformationTechnology` which has the protocol attribute set to COMBased shall be set to the length value of the `ISignalGroup` that is mapped into the `ISignalIPdu`. The `ISignalGroup` refers to the `DataTransformation` in the role `comBasedSignalGroupTransformation` which refers to a `TransformationTechnology` in the `transformerChain`.]()

The setting of V0 to the `ISignalGroup` length is motivated by `[SWS_Com_00845]` that defines that the length of the uint8-array representation equals the number of bytes which all group signals in the `ISignalGroup` occupy in the `ISignalIPdu`.

Please note that for the UINT8-array based access to signal groups the restrictions defined in `[SWS_Com_00841]` apply.

[constr_3132] Required COM Based Transformation for comBasedSignalGroupTransformation [If a `ISignalGroup` has a reference to the `DataTransformation` element in the role `comBasedSignalGroupTransformation` then this `DataTransformation` shall be the handled by the COM Based Transformer [17].]()

Note that the `SystemSignalGroup` in this case not only contains the application data element signals mapped by the `SenderReceiverToSignalGroupMapping` but also the data which has been added by the transformers (e.g. crc, sequence counter, ...).

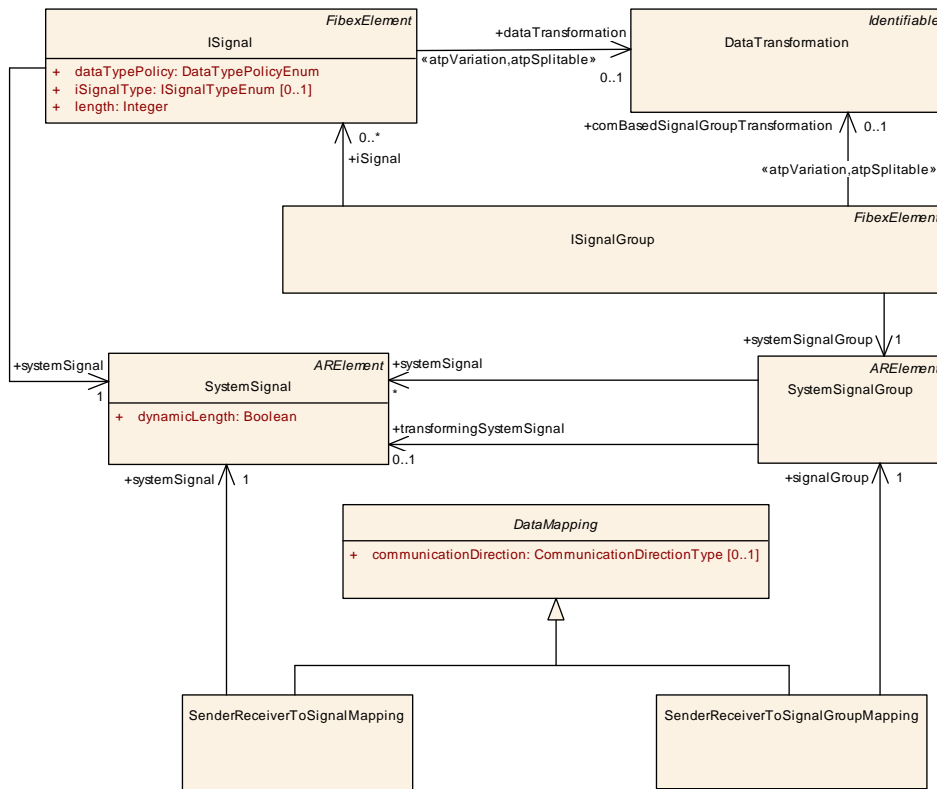


Figure 7.18: Transformer Data Mapping

[constr_3183] ISignalGroup with transformationISignalProps [An ISignalGroup that aggregates transformationISignalProps shall reference the DataTransformation in the role comBasedSignalGroupTransformation.]()

[TPS_SYST_02068] Transformer header field representation in an ISignalGroup [In case ISignalGroup has a reference to a DataTransformation in the role comBasedSignalGroupTransformation and the DataTransformation has further TransformationTechnologies defined in the role transformerChain then space for the individual headers shall be allocated by defining one ISignal per header part that is member of the ISignalGroup.](RS_SYST_00056)

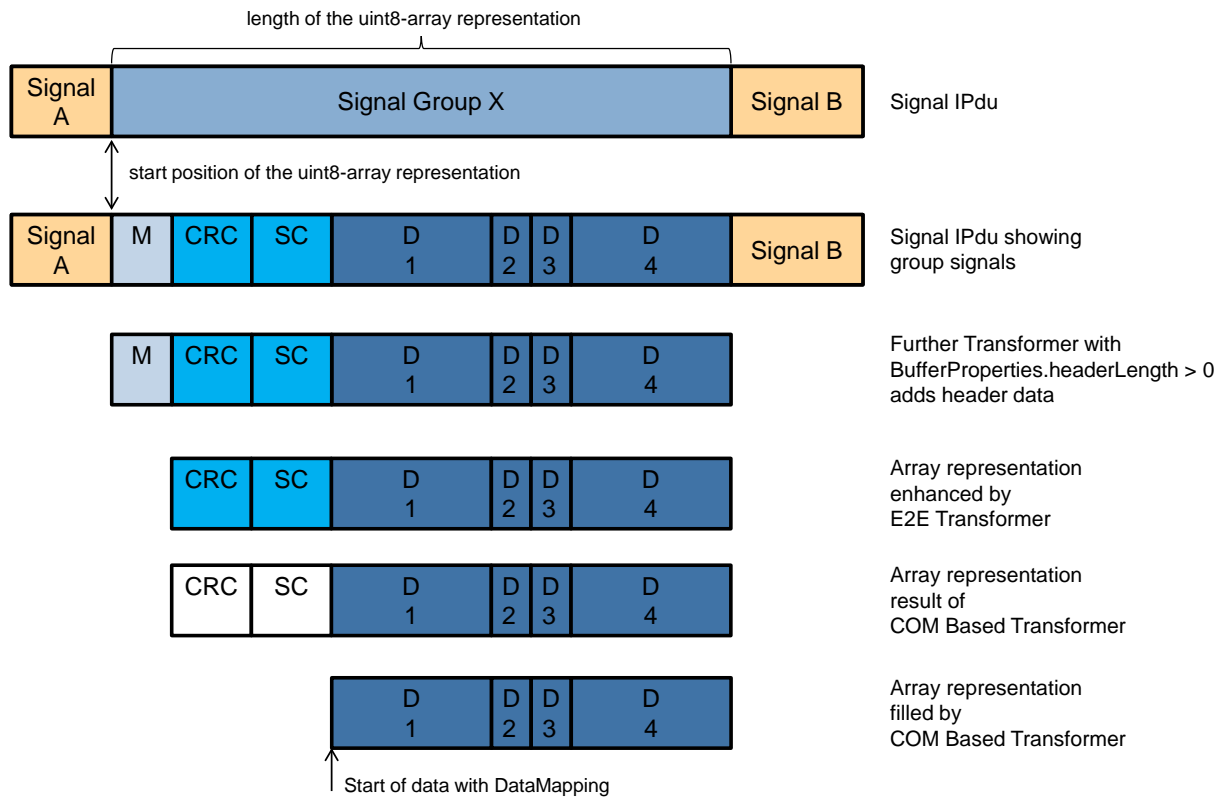


Figure 7.19: Example of COM Based Transformer buffer layout

As shown in figure 7.19 the example illustrates that for the E2E header ('CRC' and 'SC') and the 'M' header three further `ISignals` are defined within the `ISignalGroup` in order to compensate the space required by the additional transformers.

7.3.4 E2E Transformer

This section specifies the configuration of the E2E protection that is invoked "out-of-box" by RTE and realized by E2E Transformer [34], E2E Library [20] and CRC Library [35].

The specific configuration for an E2E transformer takes place in [EndToEndTransformationDescription](#) and [EndToEndTransformationISignalProps](#) shown in Figure 7.20 and in [EndToEndTransformationComSpecProps](#) (see more details in [5])

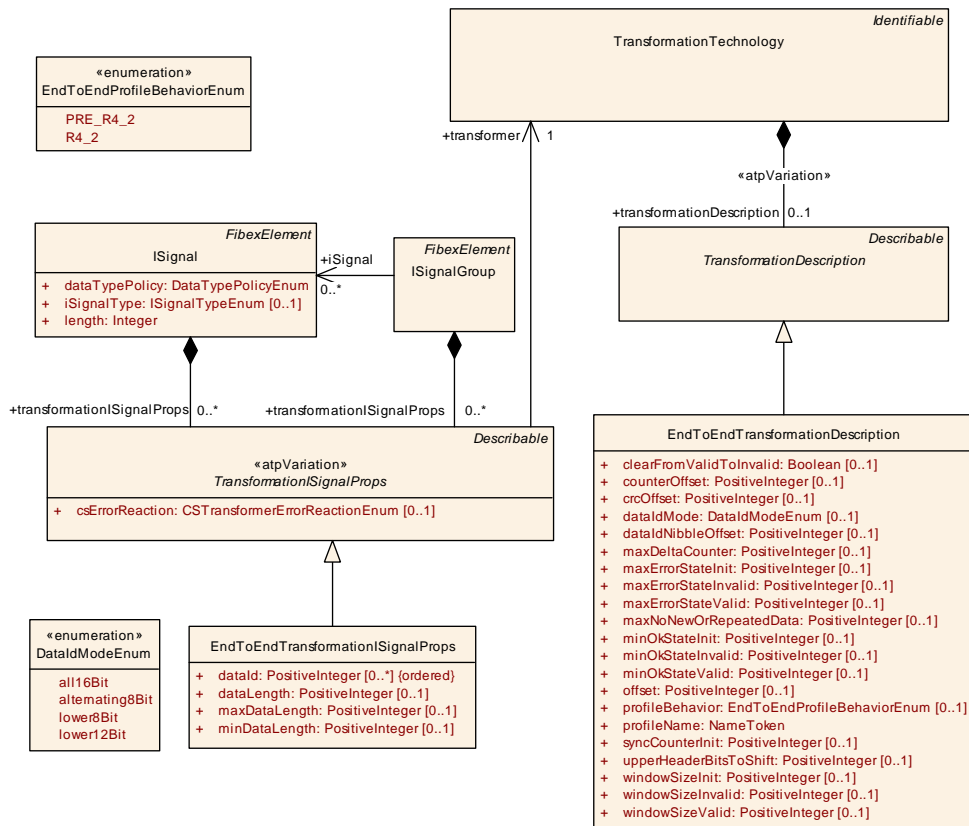


Figure 7.20: E2E Transformer Configuration

Class	EndToEndTransformationDescription			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.			
Base	<i>ARObject</i> , <i>Describable</i> , <i>TransformationDescription</i>			
Attribute	Type	Mult.	Kind	Note
clearFromValidToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
counterOffset	PositiveInteger	0..1	attr	Offset of the counter in the Data[] array in bits.





Class	EndToEndTransformationDescription			
crcOffset	PositiveInteger	0..1	attr	Offset of the CRC in the Data[] array in bits.
dataIdMode	DataIdModeEnum	0..1	attr	This attribute describes the inclusion mode that is used to include the implicit two-byte Data ID in the one-byte CRC.
dataIdNibbleOffset	PositiveInteger	0..1	attr	Offset of the Data ID nibble in the Data[] array in bits.
maxDeltaCounter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and MaxDeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorStateInit	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last WindowSize checks, for the state E2E_SM_INIT.
maxErrorStateInvalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last WindowSize checks, for the state E2E_SM_INVALID.
maxErrorStateValid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last WindowSize checks, for the state E2E_SM_VALID.
maxNoNewOrRepeatedData	PositiveInteger	0..1	attr	The maximum allowed amount of consecutive failed counter checks.
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT.
minOkStateInvalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID.
minOkStateValid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID.
offset	PositiveInteger	0..1	attr	Offset of the E2E header in the Data[] array in bits.
profileBehavior	EndToEndProfileBehaviorEnum	0..1	attr	Behavior of the check functionality
profileName	NameToken	1	attr	Definition of the E2E profile.
syncCounterInit	PositiveInteger	0..1	attr	Number of checks required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
upperHeaderBitsToShift	PositiveInteger	0..1	attr	<p>This attribute describes the number of upper-header bits to be shifted.</p> <p>value = 0 or not present: shift of upper header is NOT performed.</p> <p>value > 0: the E2E Transformer on the protect-side, takes the first upperHeaderBitsToShift bits from the upper buffer (e.g. SOME/IP header part generated by SOME/IP transformer) and shifts them towards the lower bytes and bits within the Data[] for the length of the E2E header (e.g. 12 bytes in case of E2E Profile 4). This means the shift distance is fixed - it depends on the E2E header size - what is configured here is the number of bits that are to</p>



△

Class	EndToEndTransformationDescription			
				△ be shifted. This option is defined because the Some/IP header generated by SOME/IP transformer shall be, due to compatibility between non-protected and E2E-protected communication, at the same position, which is before E2E header.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.
windowSizeInvalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSizeValid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table 7.26: EndToEndTransformationDescription

Enumeration	DataIdModeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Supported inclusion modes to include the implicit two-byte Data ID in the one-byte CRC.
Literal	Description
all16Bit	Two bytes are included in the CRC (double ID configuration). Tags: atp.EnumerationLiteralIndex=0
alternating8Bit	One of the two bytes byte is included, alternating high and low byte, depending on parity of the counter (alternating ID configuration). For even counter low byte is included; For odd counters the high byte is included. Tags: atp.EnumerationLiteralIndex=1
lower12Bit	The low byte is included in the implicit CRC calculation, the low nibble of the high byte is transmitted along with the data (i.e. it is explicitly included), the high nibble of the high byte is not used. This is applicable for the IDs up to 12 bits. Tags: atp.EnumerationLiteralIndex=2
lower8Bit	Only low byte is included, high byte is never used. This is applicable if the IDs in a particular system are 8 bits. Tags: atp.EnumerationLiteralIndex=3

Table 7.27: DataIdModeEnum

Enumeration	EndToEndProfileBehaviorEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Behavior of the check functionality
Literal	Description
PRE_R4_2	Check has the legacy behavior, before AUTOSAR Release 4.2. Tags: atp.EnumerationLiteralIndex=0
R4_2	Check behaves like new P4/P5/P6 profiles introduced in AUTOSAR Release 4.2. Tags: atp.EnumerationLiteralIndex=1

Table 7.28: EndToEndProfileBehaviorEnum

Class	<<atpVariation>> EndToEndTransformationSignalProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Holds all the ISignal specific attributes for the EndToEndTransformer.			
Base	ARObject, Describable , TransformationSignalProps			
Attribute	Type	Mult.	Kind	Note
dataId (ordered)	PositiveInteger	*	attr	This represents a unique numerical identifier. Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection.
dataLength	PositiveInteger	0..1	attr	Length of payload and E2E header in bits.
maxDataLength	PositiveInteger	0..1	attr	Maximum length of payload and E2E header in bits.
minDataLength	PositiveInteger	0..1	attr	Minimum length of payload and E2E header in bits.

Table 7.29: EndToEndTransformationSignalProps

[constr_3313] E2E transformer configuration [For each [TransformationDescription](#) variant that is a [EndToEndTransformationDescription](#)

- attribute [protocol](#) of [TransformationTechnology](#) shall be set to E2E
- attribute [version](#) of [TransformationTechnology](#) shall be set to 1.0.0
- attribute [transformerClass](#) of [TransformationTechnology](#) shall be set to `safety`

]()

[TPS_SYST_02067] E2E profile [The E2E profile is defined by [EndToEndTransformationDescription.profileName](#).] ([RS_SYST_00056](#))

[TPS_SYST_02073] [EndToEndTransformationDescription.profileName](#) [[EndToEndTransformationDescription.profileName](#) can have the following values: PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22.] ([RS_SYST_00056](#))

[TPS_SYST_02072] `profileName` of `EndToEndTransformationDescription`
[The values for the `profileName` of `EndToEndTransformationDescription` mentioned in [TPS_SYST_02073] are standardized and reserved for being used in the way the AUTOSAR standard foresees. In addition, it is positively possible to use other than the standardized values for the `profileName`.] (*RS_SYST_00056*)

The setting of the `EndToEndTransformationDescription.profileName` has an influence on the upper- and lower multiplicities of certain attributes of `EndToEndTransformationDescription` and `EndToEndTransformationISignalProps`.

[constr_3185] Multiplicity of `EndToEndTransformationDescription.dataIdMode` in PROFILE_01 and PROFILE_11 [If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.dataIdMode` attribute shall be 1.]()

[constr_3186] Multiplicity of `EndToEndTransformationDescription.dataIdMode` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 [If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 then the multiplicity of the `EndToEndTransformationDescription.dataIdMode` attribute shall be 0.]()

[constr_3326] Allowed values for `EndToEndTransformationDescription.dataIdMode` in PROFILE_11 [If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_11 then the value of the `EndToEndTransformationDescription.dataIdMode` attribute shall be set to `all16Bit` or `lower12Bit`.]()

[constr_3187] Multiplicity of `EndToEndTransformationDescription.counterOffset` in PROFILE_01 and PROFILE_11 [If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.counterOffset` attribute shall be 1.]()

[constr_3188] Multiplicity of `EndToEndTransformationDescription.counterOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 [If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 then the multiplicity of the `EndToEndTransformationDescription.counterOffset` attribute shall be 0.]()

[constr_3189] Multiplicity of `EndToEndTransformationDescription.crcOffset` in PROFILE_01 and PROFILE_11 [If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.crcOffset` attribute shall be 1.]()

[constr_3190] Multiplicity of `EndToEndTransformationDescription.crcOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07,

PROFILE_22 [If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 then the multiplicity of the `EndToEndTransformationDescription.crcOffset` attribute shall be 0.]()

[constr_3193] Multiplicity of `EndToEndTransformationDescription.offset` in PROFILE_01 and PROFILE_11 [If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.offset` attribute shall be 0.]()

[constr_3194] Multiplicity of `EndToEndTransformationDescription.offset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 [If the `EndToEndTransformationDescription.profileName` attribute is set to a value PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 then the multiplicity of the `EndToEndTransformationDescription.offset` attribute shall be 1.]()

[constr_3191] Multiplicity of `EndToEndTransformationDescription.dataIdNibbleOffset` in PROFILE_01, PROFILE_11 and `dataIdMode` equal to `lower12Bit` [If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 and the value of the `EndToEndTransformationDescription.dataIdMode` attribute is set to `lower12Bit` then the multiplicity of the `EndToEndTransformationDescription.dataIdNibbleOffset` attribute shall be 1.]()

[constr_3192] Multiplicity of `EndToEndTransformationDescription.dataIdNibbleOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 or `dataIdMode` different from `lower12Bit` [If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 or the `EndToEndTransformationDescription.dataIdMode` attribute is set to value different from `lower12Bit` then the multiplicity of the `EndToEndTransformationDescription.dataIdNibbleOffset` attribute shall be 0.]()

[constr_3148] `executeDespiteDataUnavailability` setting in case an E2E Transformer is used [A transformer chain using E2E shall be configured with `DataTransformation.executeDespiteDataUnavailability = TRUE`.]()

[constr_3149] `TransformationTechnology.needsOriginalData` settings for E2E Transformer [The `TransformationTechnology.needsOriginalData` attribute of a `TransformationTechnology` element of an E2E transformer shall be set to FALSE.]()

[constr_3151] `BufferProperties.headerLength` settings for an E2E transformer used in combination with a SOME/IP transformer [The `BufferProperties.headerLength` for an E2E transformer located in a transformer chain with a SOME/IP transformer shall be configured with the following values depending on the value of the `EndToEndTransformationDescription.profileName` attribute:

1. PROFILE_01: `BufferProperties.headerLength` = 16 bits
2. PROFILE_02: `BufferProperties.headerLength` = 16 bits
3. PROFILE_04: `BufferProperties.headerLength` = 96 bits
4. PROFILE_05: `BufferProperties.headerLength` = 24 bits
5. PROFILE_06: `BufferProperties.headerLength` = 40 bits
6. PROFILE_07: `BufferProperties.headerLength` = 160 bits
7. PROFILE_11: `BufferProperties.headerLength` = 16 bits
8. PROFILE_22: `BufferProperties.headerLength` = 16 bits

]()

This means that the E2E header in profiles 1 and 2 use 2 bytes when using SOME/IP transformer. This yields four unused bits in case of some recommended configuration settings of Profile 1 and 2. Those unused bits are set to 0xF by the E2E transformer on the sender side.

[constr_3152] `BufferProperties.headerLength` settings for an E2E transformer used in combination with a COM Based transformer [An E2E transformer used in a transformer chain with a COM Based transformer shall be configured with the following values:

- `BufferProperties.headerLength` = 0

]()

This is because the space for the E2E header needs to be allocated by a proper `ISignalGroup` layout according to [TPS_SYST_02068].

[constr_3153] E2E header field reservation required by COM Based transformer [A COM Based transformer that is used in a transformer chain with an E2E transformer requires that the following amount of space is allocated for the E2E header fields using a proper `ISignalGroup` layout according to [TPS_SYST_02068]:

PROFILE_1: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_1: if `dataIdMode` != `lower12Bit`: 12 bits

PROFILE_2: 16 bits

PROFILE_4: 96 bits

PROFILE_5: 24 bits

PROFILE_6: 40 bits

PROFILE_7: 160 bits

PROFILE_11: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_11: if `dataIdMode == all16Bit`: 12 bits

PROFILE_22: 16 bits

]()

[constr_3154] `BufferProperties.bufferComputation` setting for an E2E transformer when used together with a Com-based transformer [The `BufferProperties.bufferComputation` of an E2E transformer used in a transformer chain with a COM Based transformer shall be configured in the following way:

```
<BUFFER-COMPUTATION>
  <COMPU-RATIONAL-COEFFS>
    <COMPU-NUMERATOR>
      <V>0</V>
      <V>1</V>
    </COMPU-NUMERATOR>
    <COMPU-DENOMINATOR>
      <V>1</V>
    </COMPU-DENOMINATOR>
  </COMPU-RATIONAL-COEFFS>
</BUFFER-COMPUTATION>
```

]()

[constr_3184] Only one `EndToEndTransformationISignalProps.dataId` element in **PROFILE_01** and **PROFILE_11** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_01** or **PROFILE_11** then the multiplicity of the `EndToEndTransformationISignalProps.dataId` attribute shall be 1.]()

[constr_3156] Allowed values for `EndToEndTransformationISignalProps.dataId` in **PROFILE_01** and **PROFILE_11** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_01** or **PROFILE_11** then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 0-65535.]()

[constr_3157] Allowed values for `EndToEndTransformationISignalProps.dataId` in **PROFILE_01** and **PROFILE_11** in case `dataIdMode` is set to **lower12Bit** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_01** or **PROFILE_11** and the value of `EndToEndTransformationDescription.dataIdMode` attribute has a value of **lower12Bit** then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 256-65535.]()

[constr_3158] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in **PROFILE_01** and **PROFILE_11** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_01** or **PROFILE_11** then the attribute `maxDeltaCounter` shall be in the range 1-14.]()

[constr_3195] Allowed values for **EndToEndTransformationDescription.maxDeltaCounter** in **PROFILE_02** and **PROFILE_22** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_02** or **PROFILE_22** then the attribute **maxDeltaCounter** shall be in the range 1-15.]()

[constr_3159] Allowed values for **EndToEndTransformationDescription.maxDeltaCounter** in **PROFILE_04** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_04** the value of **maxDeltaCounter** attribute shall be in the range 1-65535.]()

[constr_3196] Allowed values for **EndToEndTransformationDescription.maxDeltaCounter** in **PROFILE_05** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_05** then the attribute **maxDeltaCounter** shall be in the range 1-255.]()

[constr_3197] Allowed values for **EndToEndTransformationDescription.maxDeltaCounter** in **PROFILE_06** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_06** then the attribute **maxDeltaCounter** shall be in the range 1-255.]()

[constr_3316] Allowed values for **EndToEndTransformationDescription.maxDeltaCounter** in **PROFILE_07** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_07** the value of **maxDeltaCounter** attribute shall be in the range 1-4'294'967'295.]()

[constr_3160] **EndToEndTransformationISignalProps.dataId** in **PROFILE_02** and **PROFILE_22** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_02** or **PROFILE_22** then the multiplicity of the **dataId** attribute shall be 16 and the value of each instance shall be in the range 0..255.]()

[constr_3161] **EndToEndTransformationISignalProps.dataLength** in **PROFILE_01**, **PROFILE_02**, **PROFILE_05**, **PROFILE_11**, **PROFILE_22** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_01**, **PROFILE_02**, **PROFILE_05**, **PROFILE_11**, or **PROFILE_22** then the multiplicity of the **EndToEndTransformationISignalProps.dataLength** attribute shall be 1.]()

[constr_3162] **EndToEndTransformationISignalProps.minDataLength** and **EndToEndTransformationISignalProps.maxDataLength** in **PROFILE_01**, **PROFILE_02**, **PROFILE_05**, **PROFILE_11**, **PROFILE_22** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_01**, **PROFILE_02**, **PROFILE_05**, **PROFILE_11**, or **PROFILE_22** then the multiplicity of the attributes **EndToEndTransformationISignalProps.minDataLength** and **EndToEndTransformationISignalProps.maxDataLength** shall be 0.]()

[constr_3163] **EndToEndTransformationISignalProps.minDataLength** and **EndToEndTransformationISignalProps.maxDataLength** in **PROFILE_04**, **PROFILE_06**, **PROFILE_07** [If the **EndToEndTransformationDescription.profileName** attribute has a value of **PROFILE_04**, **PROFILE_06**, or **PROFILE_07**

then the multiplicity of the attributes `EndToEndTransformationISignalProps.minDataLength` and `EndToEndTransformationISignalProps.maxDataLength` shall be 1.]()

[constr_3164] `EndToEndTransformationISignalProps.dataLength` in **PROFILE_04, PROFILE_06, PROFILE_07** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_04, PROFILE_06, or PROFILE_07** then the multiplicity of the attribute `EndToEndTransformationISignalProps.dataLength` shall be 0.]()

[constr_3533] `EndToEndTransformationISignalProps.dataLength` shall be a multiple of 8 [The value of `EndToEndTransformationISignalProps.dataLength`, `EndToEndTransformationISignalProps.maxDataLength`, and `EndToEndTransformationISignalProps.minDataLength` shall be a multiple of 8.]()

[constr_3155] Allowed values for `EndToEndTransformationDescription.upperHeaderBitsToShift` [The value of of the `EndToEndTransformationDescription.upperHeaderBitsToShift` attribute depends on the used serializing transformer:

COM based transformer: 0 (no bits are shifted)

SOME/IP transformer: 64 (to support the header shift of SOME/IP).

Custom transformer: no restriction (depends on header length and placement of custom transformer)

]()

[constr_3165] Effect of `EndToEndTransformationDescription.upperHeaderBitsToShift` value in **PROFILE_01, PROFILE_11** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_01 or PROFILE_11** and the serializing transformer is different than the `ComBasedTransformer` then:

1. `EndToEndTransformationDescription.crcOffset` shall be set to the same value of `upperHeaderBitsToShift`.
2. `EndToEndTransformationDescription.counterOffset` shall be set to the value of `upperHeaderBitsToShift + 8`.
3. (if used) `EndToEndTransformationDescription.dataIdNibbleOffset` shall be set to the value of `upperHeaderBitsToShift + 12`.

]()

[constr_3327] Effect of `EndToEndTransformationDescription.upperHeaderBitsToShift` value in **PROFILE_22** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_22** and the

serializing transformer is different than the ComBasedTransformer, then `EndToEndTransformationDescription.offset` shall be set to the same value of `upperHeaderBitsToShift`.`]()`

This means that the E2E header of profile 1 and profile 11, when used with SOME/IP Transformer or a Custom Transformer, is not spread across application data, but is a consecutive block of bytes. The layout flexibility available with these E2E protection profiles is therefore only supported if the ComBasedTransformer is used in combination with the E2E Transformer.

[constr_3166] `EndToEndTransformationDescription.upperHeaderBitsToShift` in **PROFILE_02** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_02** then the value of the `upperHeaderBitsToShift` attribute shall be 0.]`()`

[constr_3169] **Attribute multiplicities and values in PROFILE_02 and PROFILE_22** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_02** or **PROFILE_22** then:

1. the multiplicity of the `EndToEndTransformationDescription.crcOffset` attribute shall be 0.
2. the multiplicity of the `EndToEndTransformationDescription.counterOffset` attribute shall be 0.
3. the multiplicity of the `EndToEndTransformationDescription.dataIdNibbleOffset` attribute shall be 0.
4. the value of the `EndToEndTransformationDescription.offset` attribute shall be 0.

`]()`

[constr_3167] **Effect of `EndToEndTransformationDescription.upperHeaderBitsToShift` value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PROFILE_04**, **PROFILE_05**, **PROFILE_06**, or **PROFILE_07** the value of the `EndToEndTransformationDescription.offset` attribute shall be equal to the value of the `EndToEndTransformationDescription.upperHeaderBitsToShift` attribute.]`()`

[TPS_SYST_02194] **Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07** [If the `EndToEndTransformationDescription.profileName` attribute has a value of **PPROFILE_04**, **PROFILE_05**, **PROFILE_06**, **PROFILE_07** the E2E protected data is identified by a `EndToEndTransformationISignalProps.dataId`.]`()`

In other words if a `SystemSignal` defines the E2E protected data and a fanout is described by several `ISignals` that point to this `SystemSignal`, the `dataId` in each of those `ISignals` may have the same value.

[constr_3172] Effect of `EndToEndTransformationDescription.profileBehavior` value in PROFILE_01 [If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 and the value of the `profileBehavior` attribute is R4_2 then:

- the value of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 14.
- the value of the `EndToEndTransformationDescription.syncCounterInit` attribute shall be 1.

]()

[constr_3173] Effect of `EndToEndTransformationDescription.profileBehavior` value in PROFILE_02 [If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 and the value of the `profileBehavior` attribute is R4_2 then:

- the value of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 15.
- the value of the `EndToEndTransformationDescription.syncCounterInit` attribute shall be 1.

]()

[constr_3174] `EndToEndTransformationDescription` settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22 [If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, or PROFILE_22 then:

1. the multiplicity of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 0.
2. the multiplicity of the `EndToEndTransformationDescription.syncCounterInit` attribute shall be 0.
3. the multiplicity of the `EndToEndTransformationDescription.profileBehavior` attribute shall be 0.

]()

The `EndToEndTransformationDescription` may be differently chosen for a given `ISignal` or `ISignalGroup` depending on selected variant, with the following exceptions:

[constr_3182] Restriction on `TransformationTechnology.transformationDescriptionVariationPoint` [The `EndToEndTransformationDescription.profileName` attribute shall not be subject to variability for a given `ISignal` / `ISignalGroup`, i.e., the value of the `EndToEndTransformationDescription.profileName` attribute shall be the same in all different variants.]()

In other words, it is not possible that in one variant PROFILE_04 is used, and in another variant PROFILE_05 is used for the same [ISignal](#) or [ISignalGroup](#).

7.3.4.1 E2E state machine settings

E2E state machine settings are set in [EndToEndTransformationDescription](#) and a subset of them can be overridden in [EndToEndTransformationComSpecProps](#). The E2E state machine is described in more detail in the E2E Protocol specification [36].

Please note that the configuration of the E2E state machines with the configuration attributes available in [EndToEndTransformationDescription](#) is restricted by [constr_3176], [constr_3177], [constr_3178], [constr_3179], [constr_3180], [constr_3181] defined in [36].

7.3.4.2 E2E recommended configuration settings

This chapter provides different configuration settings for particular E2E Profiles. Please note that in future additional recommended configuration settings might be added.

7.3.4.2.1 E2E Profile 1 configuration setting C

The E2E Profile 1 configuration setting C is foreseen for CAN/FlexRay messages that are serialized by the COM based transformer and should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using `Rte_Send / Rte_Receive`.

[TPS_SYST_02069] Recommended configuration settings for E2E Profile 1 configuration setting C [The recommended configuration settings for E2E Profile 1 configuration setting C are defined in [Table 7.30](#).] ([RS_SYST_00056](#))

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_01	Profile 1
EndToEndTransformationDescription.crcOffset	0	CRC offset
EndToEndTransformationDescription.counterOffset	8	Counter offset
EndToEndTransformationDescription.dataIdNibbleOffset	12	Data Id high nibble offset
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2, i.e. one lost message.
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateInit	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSizeInvalid	3	Last 3 messages are considered

<code>EndToEndTransformationDescription.windowSizeInit</code>	2	Last 2 messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	1	One error allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	2	At least two OK messages
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	1	One error allowed
<code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>	0	no bits are shifted
<code>BufferProperties.headerLength</code>	16	16 bits is the length of E2E profile 1C header.
<code>EndToEndTransformationDescription.profileBehavior</code>	R4_2	Behavior of Profile P1 adjusted for the state machine.
<code>EndToEndTransformationDescription.maxNoNewOrRepeatedData</code>	14	Behavior of Profile P1 adjusted for the state machine.
<code>EndToEndTransformationDescription.syncCounterInit</code>	1	Behavior of Profile P1 adjusted for the state machine.

Table 7.30: Configuration of E2E Profile 1 configuration setting C

7.3.4.2.2 E2E Profile 4 configuration setting A

The E2E Profile 4 configuration setting A is foreseen for long messages that are serialized by the SOME/IP transformer. The configuration setting 4A should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using `Rte_Send` / `Rte_Receive`.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1.
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02070] Recommended configuration settings for E2E Profile 4 configuration setting A [The recommended configuration settings for E2E Profile 4 configuration setting A are defined in [Table 7.31.](#)] (*RS_SYST_00056*)

Attribute	Allowed value	Comment
<code>EndToEndTransformationDescription.profileName</code>	PROFILE_04	Profile 4
<code>EndToEndTransformationDescription.offset</code>	64	To support the fixed location of Some/IP header
<code>EndToEndTransformationDescription.maxDeltaCounter</code>	1	Maximum jump considered to be OK is 1
<code>EndToEndTransformationDescription.minOkStateInit</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateInit</code>	0	No errors allowed

<code>EndToEndTransformationDescription.windowSizeValid</code>	1	Only the last message is considered
<code>EndToEndTransformationDescription.windowSizeInvalid</code>	1	Only the last message is considered
<code>EndToEndTransformationDescription.windowSizeInit</code>	2	The two last messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	0	No errors allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	0	No errors allowed
<code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>	64	64 bits from Some/IP header to be shifted
<code>BufferProperties.headerLength</code>	96	96 bits is the length of E2E profile 4 header.

Table 7.31: Configuration of E2E configuration setting 4A

7.3.4.2.3 E2E Profile 4 configuration setting B

The E2E Profile 4 configuration setting B is foreseen for long messages that are serialized by the SOME/IP transformer. The configuration setting 4B should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using `Rte_Send` / `Rte_Receive`.

This configuration setting requires having within the monitoring window the following properties:

1. At least one OK message
2. At most one error not related to counters (e.g. CRC, data ID, length)
3. the remaining data in the monitoring window may be
 - repetitions or
 - jumps above 1.

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02071] Recommended configuration settings for E2E Profile 4 configuration setting B [The recommended configuration settings for E2E Profile 4 configuration setting B are defined in [Table 7.32.](#)] ([RS_SYST_00056](#))

Attribute	Allowed value	Comment
<code>EndToEndTransformationDescription.profileName</code>	PROFILE_04	Profile 4
<code>EndToEndTransformationDescription.offset</code>	64	To support the fixed location of Some/IP header
<code>EndToEndTransformationDescription.maxDeltaCounter</code>	2	Maximum jump considered to be OK is 2, i.e. one lost message
<code>EndToEndTransformationDescription.minOkStateInit</code>	1	At least one OK message

<code>EndToEndTransformationDescription.maxErrorStateInit</code>	1	One error allowed
<code>EndToEndTransformationDescription.windowSizeValid</code>	3	Last 3 messages are considered
<code>EndToEndTransformationDescription.windowSizeInvalid</code>	3	Last 3 messages are considered
<code>EndToEndTransformationDescription.windowSizeInit</code>	2	Last 2 messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	1	One error allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	2	At least two OK messages
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	1	One error allowed
<code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>	64	64 bits from Some/IP header to be shifted
<code>BufferProperties.headerLength</code>	96	96 bits is the length of E2E profile 4 header.

Table 7.32: Configuration of E2E Profile 4 configuration setting B

7.3.4.2.4 E2E Profile 7 configuration setting A

The E2E Profile 7 configuration setting A is foreseen for long messages (up to 4 MB) that are serialized by the SOME/IP transformer.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1.
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02134] Recommended configuration settings for E2E Profile 7 configuration setting A [The recommended configuration settings for E2E Profile 7 configuration setting A are defined in [Table 7.33.](#)] (*RS_SYST_00056*)

Attribute	Allowed value	Comment
<code>EndToEndTransformationDescription.profileName</code>	PROFILE_07	Profile 7
<code>EndToEndTransformationDescription.offset</code>	64	To support the fixed location of Some/IP header
<code>EndToEndTransformationDescription.maxDeltaCounter</code>	1	Maximum jump considered to be OK is 1
<code>EndToEndTransformationDescription.minOkStateInit</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateInit</code>	0	No errors allowed
<code>EndToEndTransformationDescription.windowSizeValid</code>	1	Only the last message is considered
<code>EndToEndTransformationDescription.windowSizeInvalid</code>	1	Only the last message is considered

<code>EndToEndTransformationDescription.windowSizeInit</code>	2	Only the last two messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	0	No errors allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	1	received message shall be OK.
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	0	No errors allowed
<code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>	64	64 bits from Some/IP header to be shifted
<code>BufferProperties.headerLength</code>	160	160 bits is the length of E2E profile 7 header.

Table 7.33: Configuration of E2E Profile 7 configuration setting A

7.3.4.2.5 E2E Profile 7 configuration setting B

The E2E Profile 7 configuration setting B is foreseen for long messages (up to 4 MB) that are serialized by the SOME/IP transformer.

This configuration setting requires having within the monitoring window the following properties:

1. At least one OK message
2. At most one error not related to counters (e.g. CRC, data ID, length)
3. the remaining data in the monitoring window may be
 - repetitions or
 - jumps above 1.

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02135] Recommended configuration settings for E2E Profile 7 configuration setting B [The recommended configuration settings for E2E Profile 7 configuration setting B are defined in [Table 7.34.](#)] ([RS_SYST_00056](#))

Attribute	Allowed value	Comment
<code>EndToEndTransformationDescription.profileName</code>	PROFILE_07	Profile 7
<code>EndToEndTransformationDescription.offset</code>	64	To support the fixed location of Some/IP header
<code>EndToEndTransformationDescription.maxDeltaCounter</code>	2	Maximum jump considered to be OK is 2, i.e. one lost message
<code>EndToEndTransformationDescription.minOkStateInit</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateInit</code>	1	One error allowed
<code>EndToEndTransformationDescription.windowSizeValid</code>	3	Last 3 messages are considered
<code>EndToEndTransformationDescription.windowSizeInvalid</code>	3	Last 3 messages are considered

<code>EndToEndTransformationDescription.windowSizeInit</code>	2	Last 2 messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	1	One error allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	2	At least two OK messages
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	1	One error allowed
<code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>	64	64 bits from Some/IP header to be shifted
<code>BufferProperties.headerLength</code>	160	160 bits is the length of E2E profile 7 header.

Table 7.34: Configuration of E2E Profile 7 configuration setting B

7.3.4.2.6 E2E Profile 11 configuration setting C

The E2E Profile 11 configuration setting C is foreseen for CAN/FlexRay messages that are serialized by the COM based transformer and should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using `Rte_Send` / `Rte_Receive`.

[TPS_SYST_02155] Recommended configuration settings for E2E Profile 11 configuration setting C [The recommended configuration settings for E2E Profile 11 configuration setting C are defined in [Table 7.35.](#)] (*RS_SYST_00056*)

Attribute	Allowed value	Comment
<code>EndToEndTransformationDescription.profileName</code>	PROFILE_11	Profile 11
<code>EndToEndTransformationDescription.crcOffset</code>	0	CRC offset
<code>EndToEndTransformationDescription.counterOffset</code>	8	Counter offset
<code>EndToEndTransformationDescription.dataIdNibbleOffset</code>	12	Data Id high nibble offset
<code>EndToEndTransformationDescription.maxDeltaCounter</code>	2	Maximum jump considered to be OK is 2, i.e. one lost message.
<code>EndToEndTransformationDescription.minOkStateInit</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateInit</code>	1	One error allowed
<code>EndToEndTransformationDescription.windowSizeValid</code>	3	Last 3 messages are considered
<code>EndToEndTransformationDescription.windowSizeInvalid</code>	3	Last 3 messages are considered
<code>EndToEndTransformationDescription.windowSizeInit</code>	2	Last 2 messages are considered
<code>EndToEndTransformationDescription.minOkStateValid</code>	1	At least one OK message
<code>EndToEndTransformationDescription.maxErrorStateValid</code>	1	One error allowed
<code>EndToEndTransformationDescription.minOkStateInvalid</code>	2	At least two OK messages
<code>EndToEndTransformationDescription.maxErrorStateInvalid</code>	1	One error allowed

<code>EndToEndTransformationDescription.upperHeader-BitsToShift</code>	0	no bits are shifted
<code>BufferProperties.headerLength</code>	16	16 bits is the length of E2E profile 1C header.

Table 7.35: Configuration of E2E Profile 11 configuration setting C

7.3.5 UserDefined Transformer

Autosar allows to describe custom Transformers that are not standardized by AUTOSAR. This is done by the usage of the following elements:

- [UserDefinedTransformationDescription](#)
- [UserDefinedTransformationISignalProps](#)
- [UserDefinedTransformationProps](#)

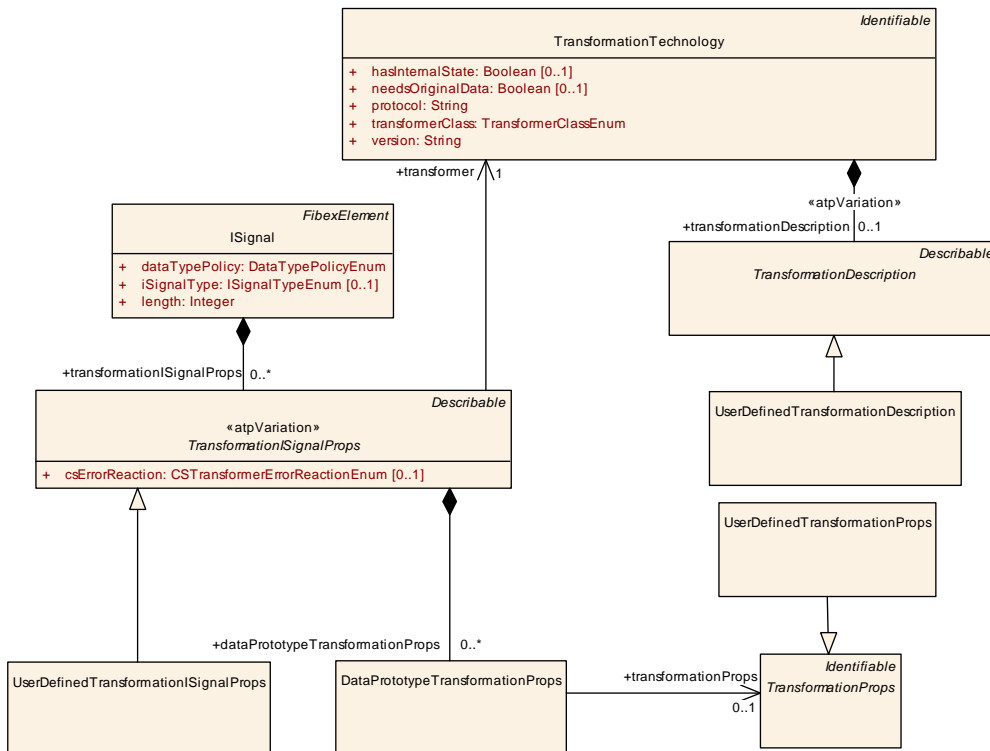


Figure 7.21: User Defined Transformation configuration

Please note that all these UserDefined classes are [Identifiable](#) or [Describable](#) and therefore are able to describe special data (sdg) which is not represented by the standard model.

Class	<<atpVariation>> UserDefinedTransformationISignalProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The UserDefinedTransformationISignalProps is used to specify ISignal specific configuration properties for custom transformers.			
Base	ARObject, Describable , TransformationISignalProps			
Attribute	Type	Mult.	Kind	Note
-	-	-	-	-

Table 7.36: UserDefinedTransformationISignalProps

Class	UserDefinedTransformationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class UserDefinedTransformationProps specifies specific configuration properties of a user defined serializer.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TransformationProps</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 7.37: UserDefinedTransformationProps

7.3.6 Support for TLV Encoding

AUTOSAR supports the usage of the so-called *Tag-Length-Value* (TLV) encoding. The following sub-sections explain the details and the extent of the support for TLV encoding.

7.3.6.1 Assignment of TLV Data Ids

[TPS_SYST_05016] Assignment of TLV data ids [The assignment of TLV data ids is done in the context of the specification of [SOMEIPTransformationISignalProps](#), namely by means of the attribute [SOMEIPTransformationISignalProps.tlvDataId.tlvDataIdDefinition.id.](#)] ([RS_SYST_00058](#))

This approach takes benefit from the fact that the [TlvDataIdDefinition](#) is able to create references to relevant model elements.

The assignment of the TLV data id is therefore done by creating such a reference and assigning a TLV data id to it by means of the attribute [TlvDataIdDefinition.id.](#)

Please note that the assignment of TLV data ids is compulsory for an entire data structure that has at least one optional member. In a nutshell, this conclusion (that is also backed by [PRS_SOMEIP_00230], see [33]) is the motivation for the existence of [[constr_1641](#)] and [[constr_1642](#)].

Please note further that the assignment of TLV data ids is not restricted to data structures with optional members. There is also a use case to support sending the elements of a specific data structure in arbitrary order even if none of the elements is considered optional.

Moreover, TLV data ids can also be assigned to arguments of a [ClientServerOperation](#). Using TLV data ids for arguments supports that arguments can be sent in arbitrary order and that new arguments can be added at arbitrary positions during the evolution of the interface. Note that optional arguments are not supported.

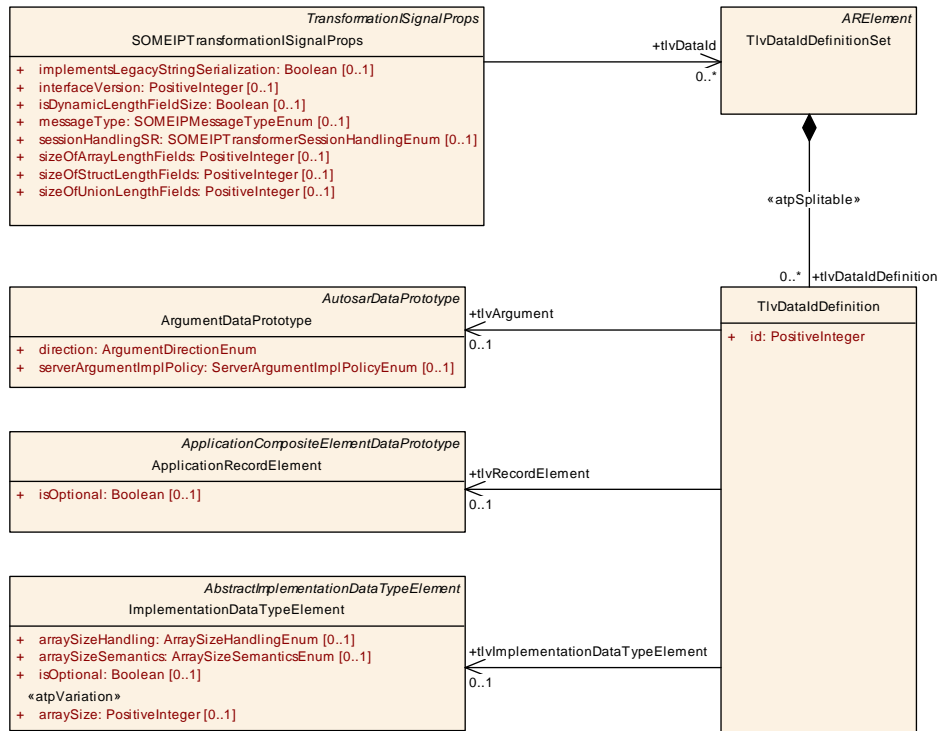


Figure 7.22: Definition of data ids for the TLV encoding inside a SOME/IP message

To sum it up: the usage of TLV data ids and optional members are two different features. Optional members require the usage of TLV data ids, but TLV data ids can also be used without having optional members.

Class	TlvDataIdDefinitionSet			
Package	M2::AUTOSARTemplates::AdaptivePlatform::ApplicationDesign::SerializationProperties			
Note	This meta-class acts as a container of TlvDataIdDefinitions to be used in a given context Tags: atp.Status=draft atp.recommendedPackage=TlvDataDefinitionSets			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
tlvDataIdDefinition	TlvDataIdDefinition	*	aggr	This aggregation represents the collection of TlvDataIdDefinitions aggregated by the TlvDataIdDefinitionSet Stereotypes: atpSplittable Tags: atp.Splitkey=tlvDataIdDefinition, tlvDataIdDefinition.id atp.Status=draft

Table 7.38: TlvDataIdDefinitionSet

Class	TlvDataIdDefinition			
Package	M2::AUTOSARTemplates::AdaptivePlatform::ApplicationDesign::SerializationProperties			
Note	This meta-class represents the ability to define the tlvDataId. Tags: atp.Status=draft			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
id	PositiveInteger	1	attr	This attribute represents the definition of the value of the TlvDataId
tlvArgument	ArgumentDataPrototype	0..1	ref	This reference assigns a tlvDataId to a given argument of a ClientServerOperation. Tags: atp.Status=draft
tlvImplementationDataTypeElement	ImplementationDataTypeElement	0..1	ref	This reference associates the definition of a TLV data id with a given ImplementationDataTypeElement. Tags: atp.Status=draft
tlvRecordElement	ApplicationRecordElement	0..1	ref	This reference associates the definition of a TLV data id with a given ApplicationRecordElement. Tags: atp.Status=draft

Table 7.39: TlvDataIdDefinition

[TPS_SYST_02211] Reference from [SOMEIPTransformationISignalProps](#) to [TlvDataIdDefinitionSet](#) [The reference from [SOMEIPTransformationISignalProps](#) to [TlvDataIdDefinitionSet](#) means that it is in the hand of the creator of a model to decide whether a global scope should be assumed or whether the definition needs to be customized for a specific case.] ([RS_SYST_00058](#))

[constr_1641] Consistent assignment of TLV data ids to [ApplicationRecordDataType](#) [For every [ApplicationRecordDataType](#) where direct members set the attribute [ApplicationRecordElement.isOptional](#) to the value `True` references to all direct members of this [ApplicationRecordDataType](#) shall be created on the basis of the definition of [TlvDataIdDefinition](#).]()

[constr_1642] Consistent assignment of TLV data ids to [ImplementationDataType](#) or [ImplementationDataTypeElement](#) [For every [ImplementationDataType](#) or [ImplementationDataTypeElement](#) of category `STRUCTURE` where direct members set the attribute [ImplementationDataTypeElement.isOptional](#) to the value `True` references to all direct members of this [ImplementationDataType](#) resp [ImplementationDataTypeElement](#) shall be created on the basis of the definition of [TlvDataIdDefinition](#).]()

The definition of a [TlvDataIdDefinition](#) that refers to an eligible model element is not limited to scenarios where optional elements are defined. It is also possible to define [TlvDataIdDefinition](#) for arbitrary methods or data structures.

A typical use case could be to prepare the argument list or subelements for future extensions. However, if one argument or sub-element is referenced then it is necessary to define references from [TlvDataIdDefinitions](#) to all other arguments or subelements as well.

[constr_1643] Completeness of the existence of a set of `TlvDataIdDefinition.tlvArguments` [If the reference `TlvDataIdDefinition.tlvArguments` exists for one `argument` of a given `ClientServerOperation` then further `TlvDataIdDefinition.tlvArguments` shall exist for all `arguments` of the given `ClientServerOperation` and all affected `TlvDataIdDefinitions` shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.]()

[constr_1644] Completeness of the existence of a set of `TlvDataIdDefinition.tlvRecordElements` [If the reference `TlvDataIdDefinition.tlvRecordElement` exists for one element of a given `ApplicationRecordDataType` then further `TlvDataIdDefinition.tlvRecordElement` shall exist for all elements of the given `ApplicationRecordDataType` and all affected `TlvDataIdDefinitions` shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.]()

[constr_1645] Completeness of the existence of a set of `TlvDataIdDefinition.tlvImplementationDataTypeElements` [Completeness of the existence of a set of `TlvDataIdDefinition.tlvImplementationDataTypeElements` If the reference `TlvDataIdDefinition.tlvImplementationDataTypeElement` exists for one `subElement` of a given `ImplementationDataType` or `ImplementationDataTypeElement` then further `TlvDataIdDefinition.tlvImplementationDataTypeElement` shall exist for all `subElements` of the given `ImplementationDataType` or `ImplementationDataTypeElement` and all affected `TlvDataIdDefinitions` shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.]()

The definition of a `TlvDataIdDefinition.id` has the purpose to provide means to unambiguously identify the argument or sub-element. For this purpose, the value of the id needs to be unique in the respective context.

[constr_1646] Scope of the uniqueness of the value of `TlvDataIdDefinition.id` for references to `ArgumentDataPrototype` [For all `TlvDataIdDefinition` that are referencing `ArgumentDataPrototypes` of a given `ClientServerOperation` in the role `tlvArgument` the attribute `TlvDataIdDefinition.id` shall exist and have a unique value in the context of respective `arguments` of the enclosing `ClientServerOperation` where attribute `direction` is set to the value `in/inout` or `out/inout`.

Note: an `argument` where attribute `direction` is set to the value `in` may have the same data id as an `argument` where attribute `direction` is set to the value `out` since the two are transferred in separate messages.]()

[constr_1647] Scope of the uniqueness of the value of `TlvDataIdDefinition.id` for references to `ApplicationRecordElement` [For all `TlvDataIdDefinition` that are referencing `ApplicationRecordElements` of a given `ApplicationDataType` in the role `tlvRecordElement` the attribute `TlvDataIdDefinition.id` shall exist and have a unique value in the context of respective enclosing `ApplicationRecordDataType`.]()

[constr_1648] Scope of the uniqueness of the value of `TlvDataIdDefinition.id` for references to `ImplementationDataTypeElement` [For all `TlvDataIdDefinition` that are referencing `ImplementationDataTypeElement`s of a given `ImplementationDataType/ImplementationDataTypeElement` in the role `tlvImplementationDataTypeElement` the attribute `TlvDataIdDefinition.id` shall exist and have a unique value in the context of respective enclosing `ImplementationDataType` or `ImplementationDataTypeElement`.]()

Obviously, it is necessary to avoid ambiguity with respect to the definition of TLV data ids. Each model element that can be assigned such an id shall only be assigned one id.

[constr_1649] `TlvDataIdDefinition` referencing `ArgumentDataPrototype` [Each `ArgumentDataPrototype` shall be referenced at most once in the role `tlvArgument` in the context of the same `SOMEIPTransformationISignalProps`.]()

[constr_1650] `TlvDataIdDefinition` referencing `ApplicationRecordElement` [Each `ApplicationRecordElement` shall be referenced at most once in the role `tlvRecordElement` in the context of the same `SOMEIPTransformationISignalProps`.]()

[constr_1651] `TlvDataIdDefinition` referencing `ImplementationDataTypeElement` [Each `ImplementationDataTypeElement` shall be referenced at most once in the role `tlvImplementationDataTypeElement` in the context of the same `SOMEIPTransformationISignalProps`.]()

As depicted in Figure 7.22, the meta-model supports the `TlvDataIdDefinition` to refer both to an `ApplicationRecordElement` as well as an `ImplementationDataTypeElement`.

In a typical case either the one or the other reference will be used and there is intentionally no constraint to explicitly use both references in a concrete model.

It would mean a significant markup in real-world AUTOSAR models to explicitly require that `TlvDataIdDefinitions` shall exist that assign concrete ids to both a given `ApplicationRecordDataType` as well as the mapped `ImplementationDataType`.

However, scenarios are conceivable that the assignment of TLV data ids may be done based on `ApplicationDataType` plus `networkRepresentationProps` on one end of the communication and based on `ImplementationDataType` on the other end.

In this case, a constraint to keep TLV data ids in sync between `ApplicationDataType` and `ImplementationDataType` would not even be helpful because either side might not know about the actual data type used as the basis of the creation of the Transformer at the other end.

Nevertheless, if both an `ApplicationDataType` and the mapped `ImplementationDataType` are annotated with TLV data ids within the same model then the associated values shall obviously be identical for corresponding sub-elements.

7.3.6.2 Assignment of TLV Wire Type

The TLV encoding supports the definition of a so-called wire type that controls how the information about the length of length fields shall be interpreted. The meaning of specific settings of the wire type is defined in [33].

[TPS_SYST_05017] Definition of the applicable wire type attribute `SOMEIPTransformationISignalProps.isDynamicLengthFieldSize` **shall be used to define the applicable wire type** [If the value of attribute `SOMEIPTransformationISignalProps.isDynamicLengthFieldSize` is set to `True` then **wire type 5-7** shall be used.

If the value of attribute `SOMEIPTransformationISignalProps.isDynamicLengthFieldSize` does not exist or is set to `False` then **wire type 4** shall be used.] (*RS_SYST_00058*)

[constr_1652] Definition of static length fields sizes in case of TLV usage [If `TlvDataIdDefinitions` are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, and `sizeOfUnionLengthFields` shall be greater than 0.]()

Rationale for the existence of [constr_1652]: The TLV serialization requires the usage of length fields. If wire type 4 is used the length field size shall be statically configured.

If wire types 5-7 (dynamic length field size) are used the static configuration of the length field size shall also be present since not all length fields are preceded by a tag, e.g. structures contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

[constr_1653] Identical values for length fields sizes in case of TLV usage [If `TlvDataIdDefinitions` are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, and `sizeOfUnionLengthFields` shall have an identical value.]()

Rationale for the existence of [constr_1653]: if an unknown member or argument is encountered the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

[constr_1654] No definition of length field sizes on DataPrototype level in case of TLV usage [If `TlvDataIdDefinitions` are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields` and `sizeOfUnionLengthFields` shall not be defined on `DataPrototype` level but only on `ISignal` level.]()

Rationale for the existence of [constr_1654]: if an unknown member or argument is encountered the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

8 Gateways

A gateway is a function within an `EcuInstance` that performs as a `FrameMapping`, `IPduMapping` or `ISignalMapping` function between two or more `CommunicationClusters`.

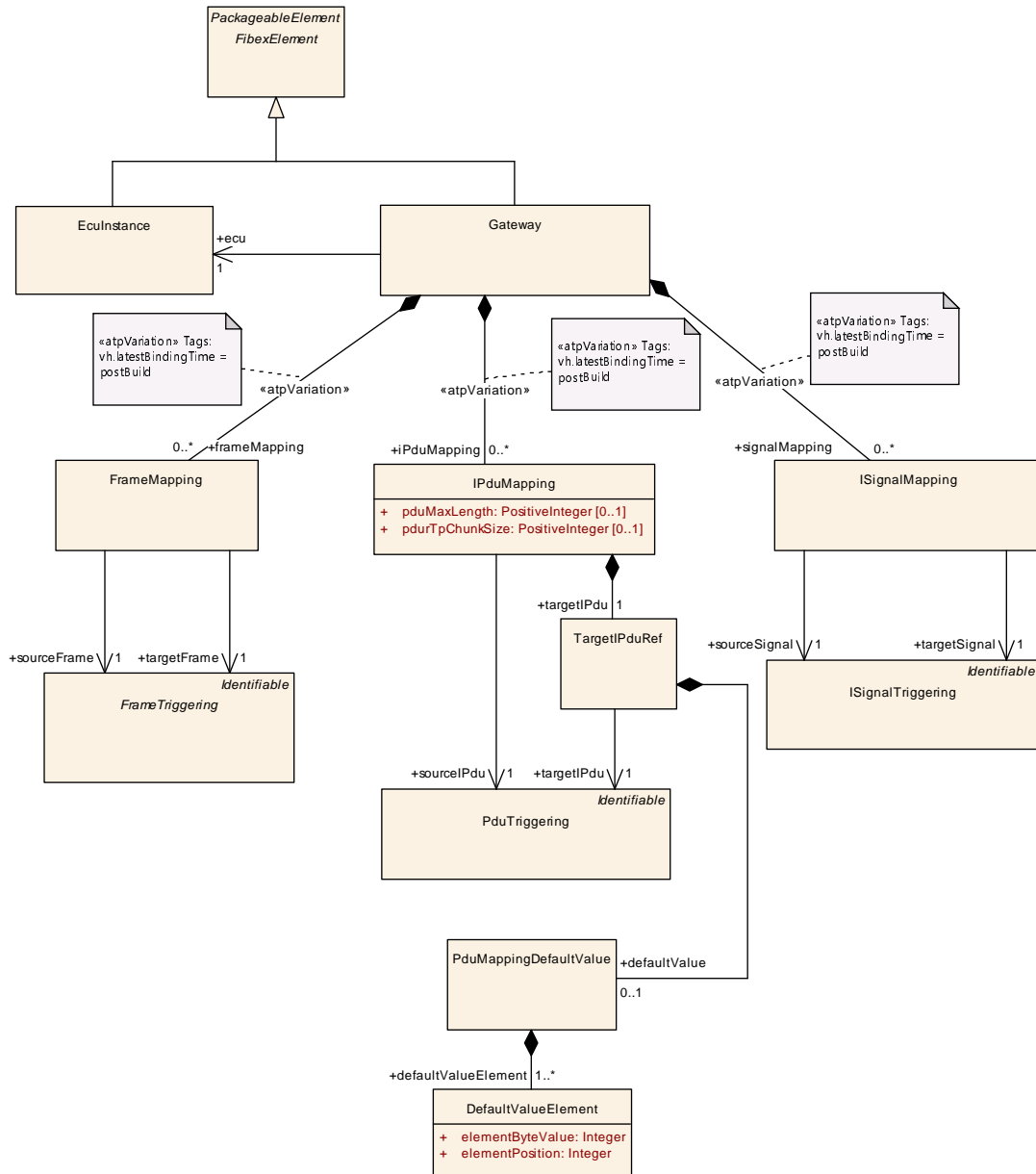


Figure 8.1: Communication Overview (Fibex4Multiplatform: Gateway)

Figure 8.1 shows the meta-model for the Gateway description in the System Template.

Class	Gateway			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	<p>A gateway is an ECU that is connected to two or more clusters (channels, but not redundant), and performs a frame, Pdu or signal mapping between them.</p> <p>Tags:atp.recommendedPackage=Gateways</p>			
Base	<p><i>ARObject</i>, <i>CollectableElement</i>, <i>FibexElement</i>, <i>Identifiable</i>, <i>MultilanguageReferrable</i>, <i>PackageableElement</i>, <i>Referrable</i></p>			
Attribute	Type	Mult.	Kind	Note
ecu	EcuInstance	1	ref	Reference to one ECU instance that implements the gateway.
frameMapping	FrameMapping	*	aggr	<p>Frame Gateway: The entire source frame is mapped as it is onto the target frame (what in general is only possible inside of a common platform). In this case source and target frame should be the identical object.</p> <p>atpVariation: If frames are variable in clusters, the gateway frame mapping needs to be variable, too.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
iPduMapping	IPduMapping	*	aggr	<p>IPdu Gateway: Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.</p> <p>atpVariation: If PDUs are variable in clusters, the gateway PDU mapping needs to be variable, too.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>
signalMapping	ISignalMapping	*	aggr	<p>Signal Gateway: Arranges those signals that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.</p> <p>atpVariation: If signals are variable in clusters, the gateway signal mapping needs to be variable, too.</p> <p>Stereotypes: atpVariation Tags:vh.latestBindingTime=postBuild</p>

Table 8.1: Gateway

8.1 Frame Mapping

The `FrameMapping` arranges those `FrameTriggerings` that are transferred by the `Gateway` from one `PhysicalChannel` to the other in pairs and defines the mapping between them. Each pair consists of a `sourceFrame` and a `targetFrame` referencing to a `FrameTriggering`.

[TPS_SYST_01116] Frame Mapping is not supported by the AUTOSAR BSW [The `FrameMapping` is not supported by the AUTOSAR BSW.] ()

The existence is optional and has been incorporated into the System Template mainly for compatibility in order to allow interchange between FIBEX and AUTOSAR descriptions.

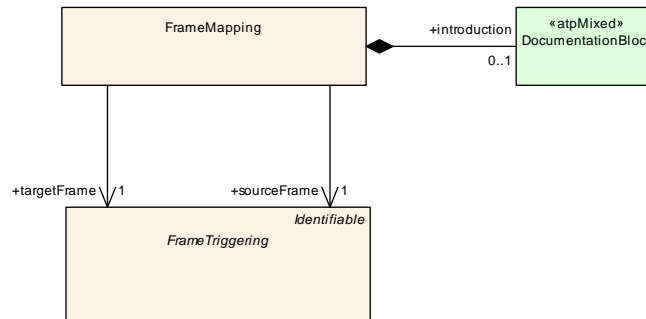


Figure 8.2: Frame Mapping (Fibex4Multiplatform: FrameMapping)

Class	FrameMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	<p>The entire source frame is mapped as it is onto the target frame (what in general is only possible inside of a common platform). In this case source and target frame should be the identical object.</p> <p>Each pair consists in a SOURCE and a TARGET referencing to a <code>FrameTriggering</code>.</p> <p>The Frame Mapping is not supported by the Autosar BSW. The existence is optional and has been incorporated into the System Template mainly for compatibility in order to allow interchange between FIBEX and AUTOSAR descriptions.</p>			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the frame mapping.
sourceFrame	FrameTriggering	1	ref	Source destination of the referencing mapping.
targetFrame	FrameTriggering	1	ref	Target destination of the referencing mapping.

Table 8.2: FrameMapping

8.2 IPdu Mapping

[TPS_SYST_01117] Pdu Gateway support [The `IPduMapping` arranges those `IPdus` that are transferred by the `Gateway` from one `PhysicalChannel` to the other (or the same) `PhysicalChannel` in pairs and defines the mapping between them. Each pair consist of a `sourceIPdu` and a `targetIPdu` referencing to a `PduTriggering`.]()

For FlexRay: If a `Pdu` is gatewayed to more than one `PhysicalChannel` of the same `CommunicationCluster`, all of this gateway relationships shall be specified. Therefore, all affected `PduTriggerings` shall be referenced in the gateway mappings.

[TPS_SYST_01118] Support of Multicast Pdu routing [The 1:n multicast routing is supported with the definition of several `IPduMappings` where the `sourceIPdu` refers to the same `PduTriggering`.]()

[TPS_SYST_02143] Support of Multisource Pdu routing [The n:1 routing is supported with the definition of several `IPduMappings` where the `targetIPdu` refers to the same `PduTriggering`.]()

Please note that in case of n:1 routing by a local module (e.g. COM, Dcm) it shall be enforced at run-time that *at most one* routing path is active (i.e., enabled via `PduR_EnableRouting()`). In case of n:1 routing by a pure gateway routing (either TP or IF) all routing paths can be active at run time.

[TPS_SYST_02207] Routing on the fly [If routing on the fly is not possible in case:

- there is more than one destination routing path or
- the routing path uses the Interface (IF) API instead of Transport Protocol (TP) API

then `IPduMapping.pdurTpChunkSize` will be ignored in the Ecu configuration.]()

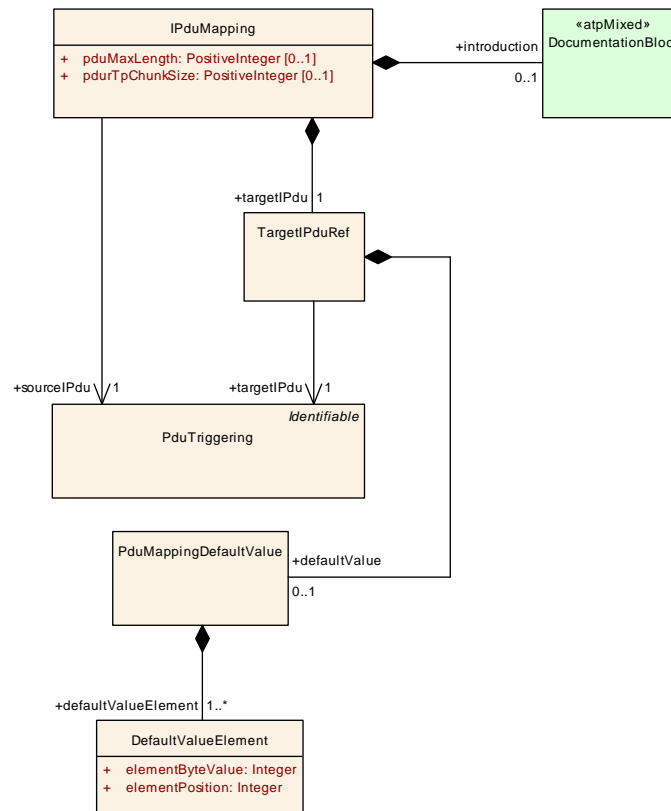


Figure 8.3: I-Pdu Mapping (Fibex4Multiplatform: IPduMapping)

Please note that the attribute `pduMaxLength` can be used to configure a maximum length which shall be forwarded to the destination module by the PduR, if the length of the received Pdu (`sourceIPdu`) exceeds `IPduMapping.pduMaxLength`. If the length of the received Pdu (`sourceIPdu`) does not exceed `IPduMapping.pduMaxLength` or the attribute is not defined, PduR shall forward the length of the received Pdu (`sourceIPdu`).

Example: Pdu A is transmitted on CAN as CAN-FD frame and forwarded by a gateway as `ContainerIPdu` to another CAN.

Pdu A in the transmitting ECU has the following configuration: `Pdu.length = 9 bytes`.

The routing path to route Pdu A as `ContainerIPdu` to another CAN has the following configuration: `sourceIPdu.length = 9 bytes`
`targetIPdu.length = 60 bytes (due to pduMaxLength = 60 bytes)`.

The gateway has no data length check configured.

1. ECU A transmits PDU A within a CAN-L-PDU with a length of 12 byte (9 byte + 3 byte padding).
2. ECU B (gateway) receives the CAN-L-PDU and CanIf forwards 12 byte to the upper layer as received length.

3. Communication matrix of ECU A is updated and payload of PDU A is extended to 60 Bytes. ECU B (gateway) is not updated.
4. ECU A transmits the extended PDU A within a CAN-L-PDU with a length of 64 byte (60 byte payload + 4 byte padding).
5. ECU B (gateway) receives the CAN-L-PDU, CanIf forwards 64 byte to PduR. PduR forwards 60 byte as received length to the IpduM. IPduM packs the received [Pdu](#) to the configured [ContainerIPdu](#).

Class	IPduMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the IPdu mapping.
pduMaxLength	PositiveInteger	0..1	attr	Define the maximum length in bytes of a Pdu which should be used as length for the corresponding source IPdu and targetIPdu. If IPduMapping.pduMaxLength is defined, the length for the corresponding sourceIPdu and targetIPdu shall be set to IPduMapping.pduMaxLength.
pduTpChunk Size	PositiveInteger	0..1	attr	Optionally defines the to be configured Pdu Router Tp ChunkSize for this routing relation.
sourceIPdu	PduTriggering	1	ref	Source destination of the referencing mapping.
targetIPdu	TargetIPduRef	1	aggr	Target destination of the referencing mapping.

Table 8.3: IPduMapping

Class	TargetIPduRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Target destination of the referencing mapping.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
defaultValue	PduMappingDefault Value	0..1	aggr	If no I-Pdu has been received a default value will be distributed.
targetIPdu	PduTriggering	1	ref	IPdu Reference

Table 8.4: TargetIPduRef

Class	PduMappingDefaultValue			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Default Value which will be distributed if no I-Pdu has been received since last sending.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note





Class	PduMappingDefaultValue			
defaultValueElement	DefaultValueElement	1..*	aggr	The default value consists of a number of elements. Each default value element is represented by the element and the position in an array.

Table 8.5: PduMappingDefaultValue

Class	DefaultValueElement			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
elementByteValue	Integer	1	attr	The integer value of a freely defined data byte.
elementPosition	Integer	1	attr	This attribute specifies the byte position of the element within the default value

Table 8.6: DefaultValueElement

8.2.1 Routing and processing of Diagnostics Pdus

An [EcuInstance](#) routes a source [DcmIPdu](#) to a destination [DcmIPdu](#) if there is an [IPduMapping](#) in place that is configured according to [TPS_SYST_01117]. The [EcuInstance](#) also processes the [DcmIPdu](#) locally if the source [DcmIPdu](#) is assigned a functional destination address.

8.3 Signal Mapping

[TPS_SYST_01119] Signal Gateway support [The `ISignalMapping` defines the mapping between `ISignals` and `ISignalGroups` that are transferred by the `Gateway` from one `PhysicalChannel` to the other (or the same) `PhysicalChannel`. Each mapping pair consists of a `sourceSignal` and a `targetSignal` referencing an `ISignalTriggering`. Each `ISignalTriggering` points to either an `ISignal` or an `ISignalGroup`. The `ISignal` refers to the to be routed `SystemSignal`, the `ISignalGroup` refers to the to be routed `SystemSignalGroup`.]()

[constr_3051] Restriction of `ISignalMapping` references [If the `sourceSignal` references an `ISignal` then the `targetSignal` shall also reference an `ISignal`.]()

[TPS_SYST_01155] Routing of `ISignalGroups` [If the `sourceSignal` references an `ISignalGroup` then the `targetSignal` can reference either an `ISignalGroup` or an `ISignal`.]()

[constr_3052] Complete `ISignalMapping` of `ISignalGroup` signals [If an `ISignalMapping` to an `ISignal` that is a member of a `ISignalGroup` exists then (see [TPS_SYST_01120]) an `ISignalMapping` to the enclosing `ISignalGroup` shall exist as well.]()

[TPS_SYST_02162] Routing of `ISignals` of `ISignalGroups` [When performing a signal group routing two approaches are supported for the pairing of the included `ISignals`:

- implicit mapping: the `ISignalMapping` points in the `sourceSignal` role to an `ISignalTriggering` of an `ISignalGroup` and no `ISignalMappings` are defined for the included `ISignals`. Identical `shortNames` of `ISignal` elements identify correlating `ISignals` between the source and the target in the scope of the `ISignalMapping`.
- explicit mapping: the `ISignalMapping` points in the `sourceSignal` role to an `ISignalTriggering` of an `ISignalGroup` and in addition explicitly specified `ISignalMappings` define which `ISignals` correlate to each other.

]()

Please note that SWS_COM [21] does not support the “implicit mapping” of [TPS_SYST_02162]. Thus it is required in the upstream mapping to derive individual `ISignalMappings` for all the members of a to be routed `ISignalGroup`.

[TPS_SYST_01120] Precedence of `ISignalMappings` [If a dedicated `ISignalMapping` for at least one `ISignal` within an `ISignalGroup` exists the implicit mapping on the basis of `shortNames` is no longer applicable for any `ISignal` within that `ISignalGroup`.]()

[TPS_SYST_01121] Support of Multicast signal routing [The 1:n multicast routing is supported with the definition of several `ISignalMappings`. See also the COM Signal Gateway fan-out description in [TPS_SYST_01110].]()

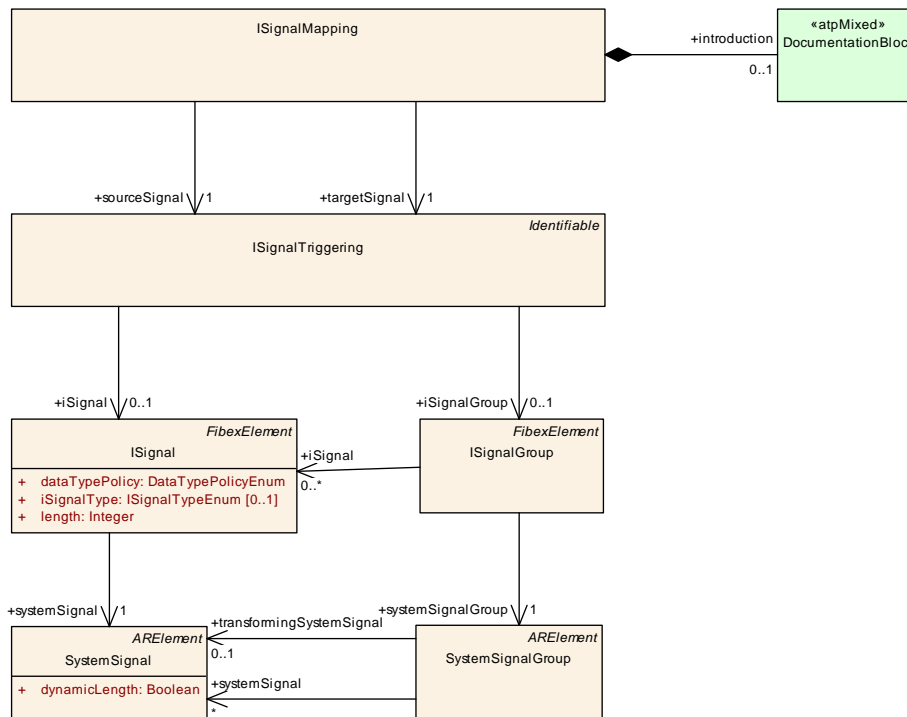


Figure 8.4: Signal Mapping (Fibex4Multiplatform: Signal Mapping)

Class	ISignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Arranges those signals (or SignalGroups) that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a source and a target referencing to a ISignalTriggering.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ISignal mapping.
sourceSignal	ISignalTriggering	1	ref	Source destination of the referencing mapping.
targetSignal	ISignalTriggering	1	ref	Target destination of the referencing mapping.

Table 8.7: ISignalMapping

8.3.1 Partial Signal Group Mapping

[TPS_SYST_01122] partial routing between ISignalGroups [The ISignalMapping supports partial routing between ISignalGroups which have not identical set of ISignals within an ISignalGroup.]()

[constr_3053] Complete ISignalMapping of target ISignalGroup [If an ISignalGroup is referenced by a targetSignal then [TPS_SYST_02162] applies for each of the contained ISignal of that ISignalGroup.]()

Figure 8.5 shows an example for a partial signal group mapping with explicit mappings for the GroupSignals.

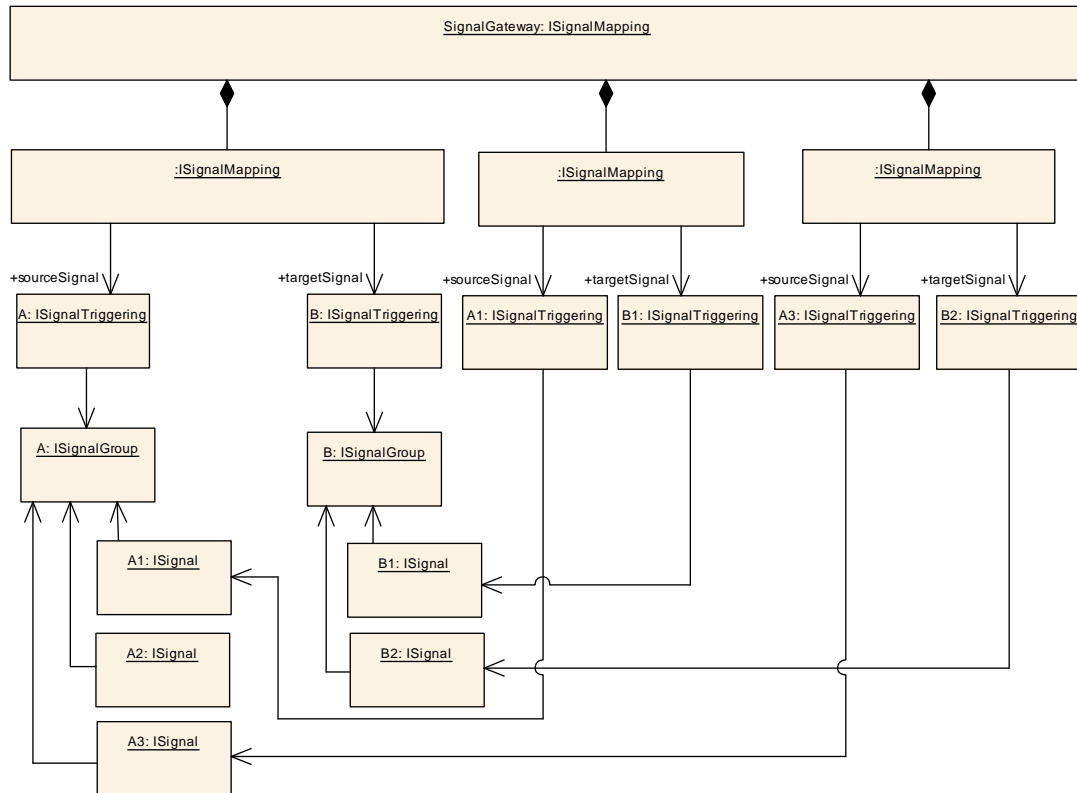


Figure 8.5: Partial Signal Group Mapping Example

9 Global Time Synchronization

9.1 Introduction

This chapter describes the modeling of how a global time synchronization in an AUTOSAR system can be achieved. There are two kinds of time bases: synchronized time base and offset time base. This manifests in two possible values for the attribute category (see [constr_3519]).

[constr_3519] Value of `category` of `GlobalTimeDomain` [The attribute `category` of `GlobalTimeDomain` can have the following values:

- SYNCHRONIZED: this time base does not depend on the existence of another time base
- OFFSET: this time base depends on the existence of another time base. It delivers a value that represents an offset relative to the referenced (`GlobalTimeDomain.offsetTimeDomain`) synchronized time base.

]()

There are several use cases for implementing a system-wide global time in an vehicle:

- In case of an accident it may be necessary to post-mortem analyze whether the vehicle ECUs performed according to specification. This implies that it shall be possible to unambiguously determine the sequence of activities before a crash. This sequence can only be determined if all components in the distributed system depend on a reliable global time basis.
- It may be necessary that several ECUs in the distributed system need to act in concert with respect to the time that a specific activity is executed. A very trivial example for this requirement is the activation of turn indicators in a car. These are rarely connected to a single ECU (which could take care of synchronously flashing the turn indicators) but their synchronized execution is still very essential for the vehicle operation.
- The distribution of several global time bases shall be possible (e.g. a vehicle local time based on the runtime of the car and a GPS-based time).
- It shall be possible to define offset time bases which have the property that they are based on a synchronized time base and distribute the offset time value as difference to the synchronized time base.

It is obvious that the distribution of global time within a vehicle requires a system-wide context and therefore, the AUTOSAR System Template defines relevant meta-classes and their relations for this purpose.

Of course, the actual implementation of global time distribution is done in a couple of basic-software modules that need to be configured in the context of integrating a particular ECU. The purpose of the meta-model described in chapter 9 is to support the configuration of these basic-software modules.

The modeling of how the distribution of global time is supposed to work can roughly be distributed into two parts, the discussion of the *big picture* (see 9.2) and the description of the details that eventually will support the configuration of the corresponding basic-software modules. The latter can be found in chapter 9.3.

9.2 The big Picture

The central part of the formalization of global time synchronization is the existence of a *global time domain*, formalized as `GlobalTimeDomain`.

However, the fragment *global* in *global time domain* primarily stresses the fact that it is supposed to support the distribution of a *global* time rather than implying an information about the scope or visibility of a `GlobalTimeDomain`¹.

In other words, there is typically more than a single `GlobalTimeDomain` available in the `System`.

Please note that the concept of the `GlobalTimeDomain` roughly corresponds to the existence of a `CommunicationCluster`, i.e. it takes at least one `CommunicationCluster` to implement a *global time domain*.

[TPS_SYST_05006] Chaining of `GlobalTimeDomains` [It is possible to extend the *global time domain* to several `CommunicationClusters` that are interconnected by means of a `Gateway`.

In other words, the global time base is routed from one `CommunicationCluster` to another, whereas the Time Slave resp. Time Slave Port updates its local time base by using the received global time base and takes into account, whether a time base correction has to be considered or not.

There are certainly use-cases for implementing a `GlobalTimeDomain` that extends to several `CommunicationClusters`, but in many (if not in the majority of) cases it will be necessary to update the time information for the sake of precision.

In this case, however, two separate `GlobalTimeDomains` rather than a single `GlobalTimeDomain` exist. The `GlobalTimeDomain` relate to each other such that one `GlobalTimeDomain` refers to the other in the role `globalTimeSubDomain`.]()

In order to understand the way how `GlobalTimeDomains` refer to each other, it is important to understand that the concept of a *global time domain* has an underlying asymmetric approach of how the time information is distributed.

That is, not all participants in the communication of global time information are able and/or entitled to update the time information and send it around for others to consume.

[TPS_SYST_02103] Semantics of `GlobalTimeDomain.domainId` [`GlobalTimeDomain.domainId` represents a specific time source, e.g. GPS time.]()

¹For the intents and purposes of this chapter, always make sure to read **global-time domain** rather than **global time-domain**.

The modeling of `GlobalTimeDomains` and `SubDomains` describes the propagation of time values of a time source through the networks. Since the specific time source corresponds to the value of `GlobalTimeDomain.domainId` [`constr_3251`] is formulated.

[constr_3251] Value of `GlobalTimeDomain.domainId` in `globalTimeSubDomain` chains [In a chain of `GlobalTimeDomain.globalTimeSubDomain` the value of the attribute `GlobalTimeDomain.domainId` shall be identical.]()

[TPS_SYST_05007] separation of roles within a `GlobalTimeDomain` [Within a single *global time domain*, There is a strict separation of roles into a single *global time master* (formalized by the meta-class `GlobalTimeMaster`) and a collection of so-called *global time slaves* (formalized by means of the meta-class `GlobalTimeSlave`).

The role of the `GlobalTimeMaster` is to provide the global time information and the role of the collection of `GlobalTimeSlaves` is to consume the information. The chaining of `GlobalTimeDomains` needs to be understood as the intention to implement the following information flow:

1. from the `GlobalTimeMaster` of one `GlobalTimeDomain` to the `GlobalTimeSlaves` of the same `GlobalTimeDomain`
2. via the `GlobalTimeMaster` of the `GlobalTimeDomain` referenced in the role `globalTimeSubDomain` to the `GlobalTimeSlaves` of the `globalTimeSubDomain`

]()

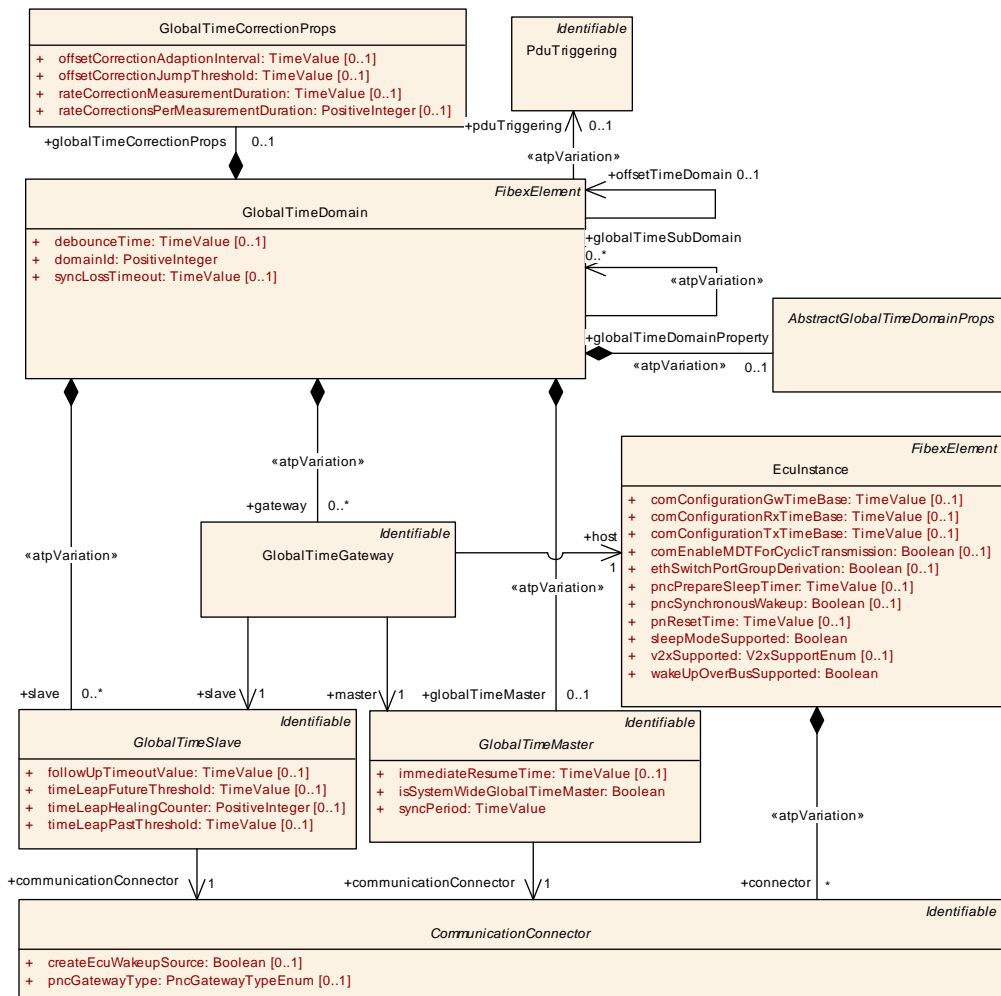


Figure 9.1: Big Picture of AUTOSAR global time synchronization

[TPS_SYST_05008] Semantics of a **GlobalTimeGateway** [In order to achieve the flow of information between a **GlobalTimeSlave** of a given **GlobalTimeDomain** to the **GlobalTimeMaster** of another **GlobalTimeDomain**, it is necessary to establish the existence of a so-called **GlobalTimeGateway**.

In terms of functionality, a **GlobalTimeGateway** complements the functionality of the underlying **Gateway** such that, on top of the mere routing from one **CommunicationCluster** to another, the time information is actively updated in the process of passing it from one **GlobalTimeDomain** to the other.]()

[TPS_SYST_05009] **GlobalTimeDomain.pduTriggering** for transmitting global time information [The flow of global time information is unidirectional, i.e. the **GlobalTimeSlaves** consume the information without providing any form of feedback to the corresponding **GlobalTimeMaster**.

Thanks to this conceptual detail, there is only the need for **one** dedicated **Pdu** for the transmission of the actual global time information in the context of one **GlobalTimeDomain**.

The characteristics of accessing the information contained in this `Pdu` do make any requirements on the nature of the `Pdu`. Therefore, it is sufficient and applicable to use the `GeneralPurposePdu` for this use case.

To make this possible, it is necessary to include the global time use case in the set of standardized values of the attribute `GeneralPurposePdu.category`. In other words, `[constr_3081]` applies. `()`

[constr_3261] `GlobalTimeDomain.pduTriggering` category [The `Pdu` that is referenced by the `PduTriggering` that in turn is referenced by `GlobalTimeDomain` in the role `pduTriggering` shall be a `GeneralPurposePdu` of category `GLOBAL_TIME`.] `()`

[TPS_SYST_05010] `GlobalTimeDomain.pduTriggering` is not required on Ethernet [The `Pdu` for transmitting global time information is not required on the Ethernet bus. Here, the information is accessed directly from the Ethernet Interfaces, i.e. the hardware already keeps track of the global time.] `()`

[constr_1369] `CommunicationConnectors` shall be attached to the same `CommunicationCluster` [All `CommunicationConnectors` referenced from `GlobalTimeMaster` and `GlobalTimeSlaves` aggregated in one `GlobalTimeDomain` shall be referenced in the role `commConnector` by the same `PhysicalChannel` aggregated by the same `CommunicationCluster`.] `()`

[constr_1370] Consistency of `GlobalTimeDomain` [The `GlobalTimeSlave` referenced in the role `GlobalTimeGateway.slave` and the `GlobalTimeMaster` referenced in the role `GlobalTimeGateway.master` shall **not** be aggregated by the same `GlobalTimeDomain`.] `()`

The background of `[constr_1370]` is that the `GlobalTimeGateway` is supposed to connect two `GlobalTimeDomains` it is hardly possible that the `GlobalTimeGateway.slave` and the `GlobalTimeMaster` can be aggregated by the same `GlobalTimeDomain`.

[TPS_SYST_05011] Ownership of `GlobalTimeGateway` [Since the existence of a `GlobalTimeGateway` is only justified if a `GlobalTimeDomain` exists that is referenced by a `GlobalTimeDomain` in the role `globalTimeSubDomain` it seems appropriate to aggregate the `GlobalTimeGateway` at the `GlobalTimeDomain` referenced in the role `globalTimeSubDomain`.] `()`

In other words, the `GlobalTimeGateway` shall be aggregated at the `GlobalTimeDomain` that also aggregates the `master`.

Please note that `GlobalTimeDomain.gateway` effectively has a 0..1 multiplicity since no more than one `globalTimeMaster` is allowed per `GlobalTimeDomain`.

[constr_1371] Consistency of attribute `host` [Within the context of an aggregating `GlobalTimeDomain`, the `CommunicationConnectors` referenced in the role `GlobalTimeGateway.master.communicationConnector` and `GlobalTimeGateway.slave.communicationConnector` shall be aggregated by the same `EcuInstance` that is referenced in the role `GlobalTimeGateway.host`.] `()`

[constr_1372] Consistency of attribute pduTriggering [Within the context of an aggregating `GlobalTimeDomain`, the `pduTriggering` shall be owned by `PhysicalChannel` that is also referencing the `CommunicationConnectors` referenced in the roles `GlobalTimeSlave.communicationConnector` and `GlobalTimeMaster.communicationConnector`.]()

[TPS_SYST_05013] Semantics of GlobalTimeMaster.isSystemWideGlobalTimeMaster [The attribute `GlobalTimeMaster.isSystemWideGlobalTimeMaster` indicates whether a given `GlobalTimeMaster` is considered an independent (i.e. **[constr_1373]** applies) source of global time information.]()

[constr_1373] GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE [`GlobalTimeMaster` with attribute `isSystemWideGlobalTimeMaster` set to `TRUE` shall not be referenced in the role `GlobalTimeGateway.master`.]()

[TPS_SYST_05014] GlobalTimeMaster.isSystemWideGlobalTimeMaster [There is no limitation regarding the number of `GlobalTimeMasters` that have attribute `isSystemWideGlobalTimeMaster` set to `TRUE`. The attribute does not imply that there can only be one `GlobalTimeMaster` within the context of a `System`.]()

[constr_1374] Only fan-out possible for GlobalTimeGateway [For all `GlobalTimeGateways` that refer to the same `EcuInstance` the condition applies that no two `GlobalTimeGateways` shall refer to the same `GlobalTimeMaster`.]()

In other words, a fan-in of time information such that time information is received from several sources is not supported.

In figure 9.2 an example of a Global Time Sync setup is shown. The *Global time master* ECU creates the *TimeDomain* and provides it to several `globalTimeSubDomains`. The `GlobalTimeMasters` for the `globalTimeSubDomains` take the *TimeDomain* and distribute it to their networks.

The time for the `GlobalTimeMasters` *TM11* and *TM21* is based on the *TimeDomain* and therefore they have the attribute `isSystemWideGlobalTimeMaster` set to *true*.

The time for the `GlobalTimeMasters` *TM12* and *TM22* are based on a `GlobalTimeGateway` and therefore they have the attribute `isSystemWideGlobalTimeMaster` set to *false*.

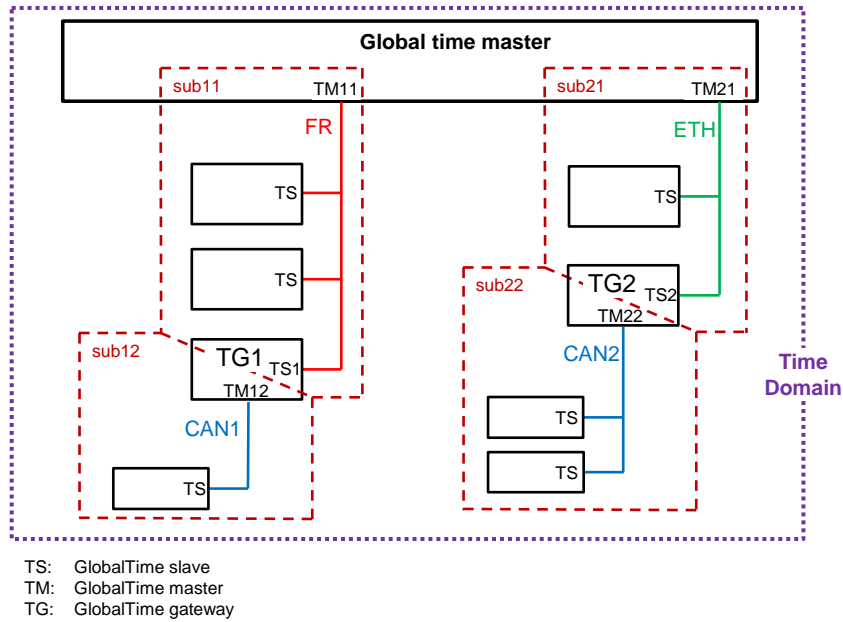


Figure 9.2: Example Global Time Sync topology

A partial outline of the example system description structure is shown in figure 9.3.

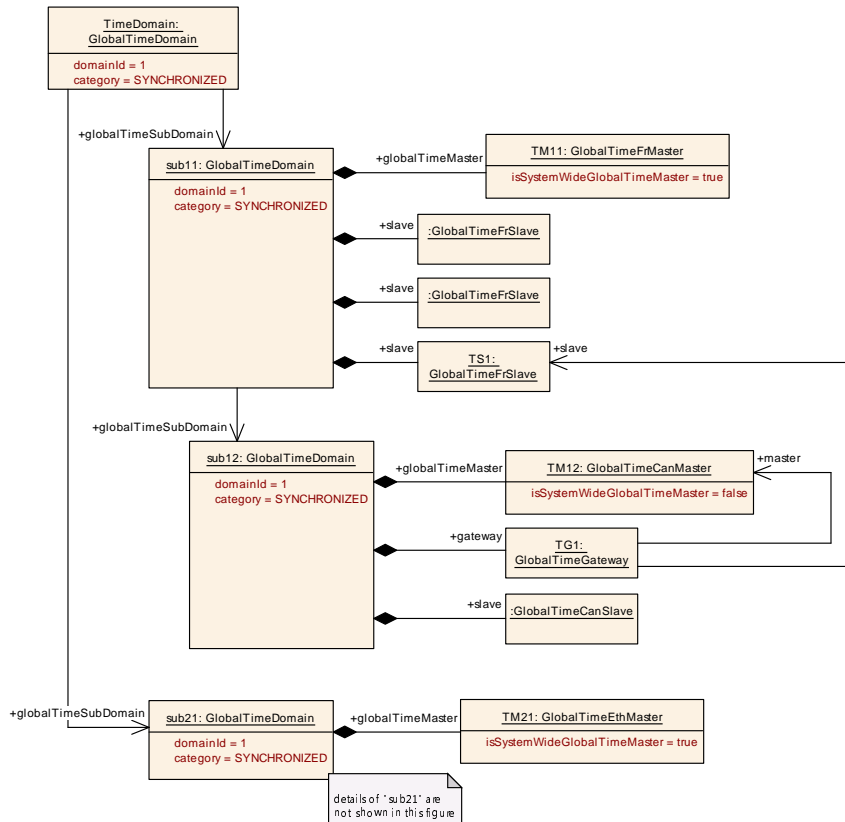


Figure 9.3: System Description of Global Time Sync example

An offset time domain is defined by an reference from a `GlobalTimeDomain` to another `GlobalTimeDomain` in the role `GlobalTimeDomain.offsetTimeDomain`.

This makes the reference source the offset time domain and the reference target the synchronized time domain.

[constr_3520] Offset time domain shall be based on a synchronized time domain
 [If a `GlobalTimeDomain` has a reference with the role `GlobalTimeDomain.offsetTimeDomain` the reference source shall have a `GlobalTimeDomain.domainId` in the range of 16-31 and the reference target shall have a `GlobalTimeDomain.domainId` in the range of 0-15.]()

Rationale: In the [37] Specification the ranges are fixed for synchronized and offset time domains.

Note that the same synchronized time domain can be referenced by several different offset time domains.

[TPS_SYST_03015] Offset time domain requires synchronized time domain
 [Since the calculation of the actual offset time domain time requires the presence of the synchronized time domain as well as the offset time domain it is required that every ECU which receives an offset time domain also receives the respective synchronized time domain.]()

In figure 9.4, an example of a Offset Time Sync setup is shown. The example is based on the setup shown in figure 9.2 and extends this with the definition of an offset time domain.

The *Global time master* ECU creates the synchronized *TimeDomain* and provides it to several `globalTimeSubDomains`. The figure needs to be interpreted in the way that the *OffsetTimeDomain* is based on the *TimeDomain* and is sort of overlaid, although drawn side by side.

The time slave *TS_O* receives the *TimeDomain* as a `GlobalTimeSlave` and also provides the *OffsetTimeDomain* as *TM51* in the role of a `GlobalTimeMaster` on the same network *FR*.

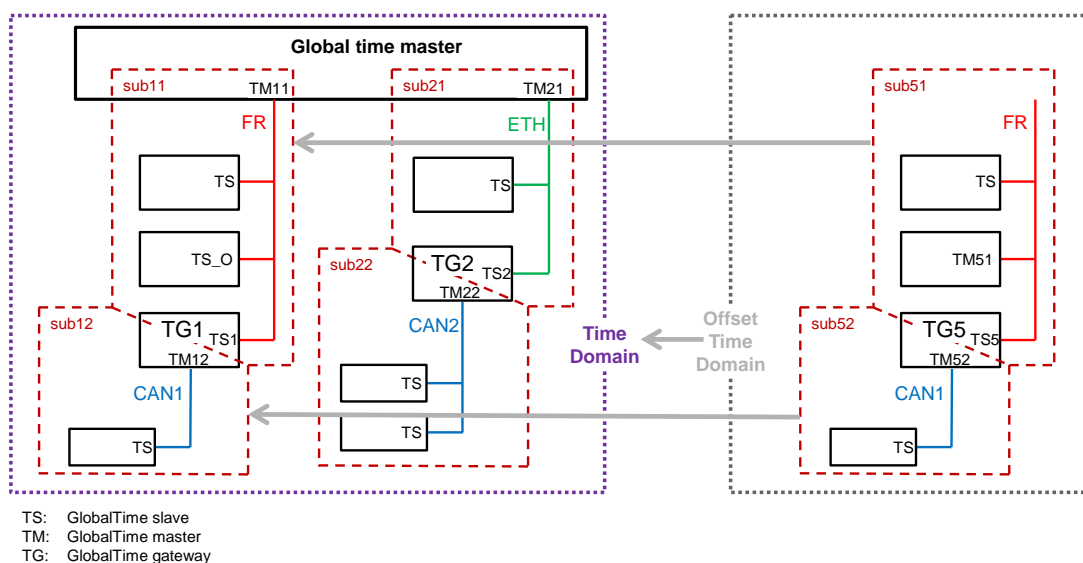


Figure 9.4: Example Offset Time Sync topology

A partial outline of the example system description structure is shown in figure 9.5.

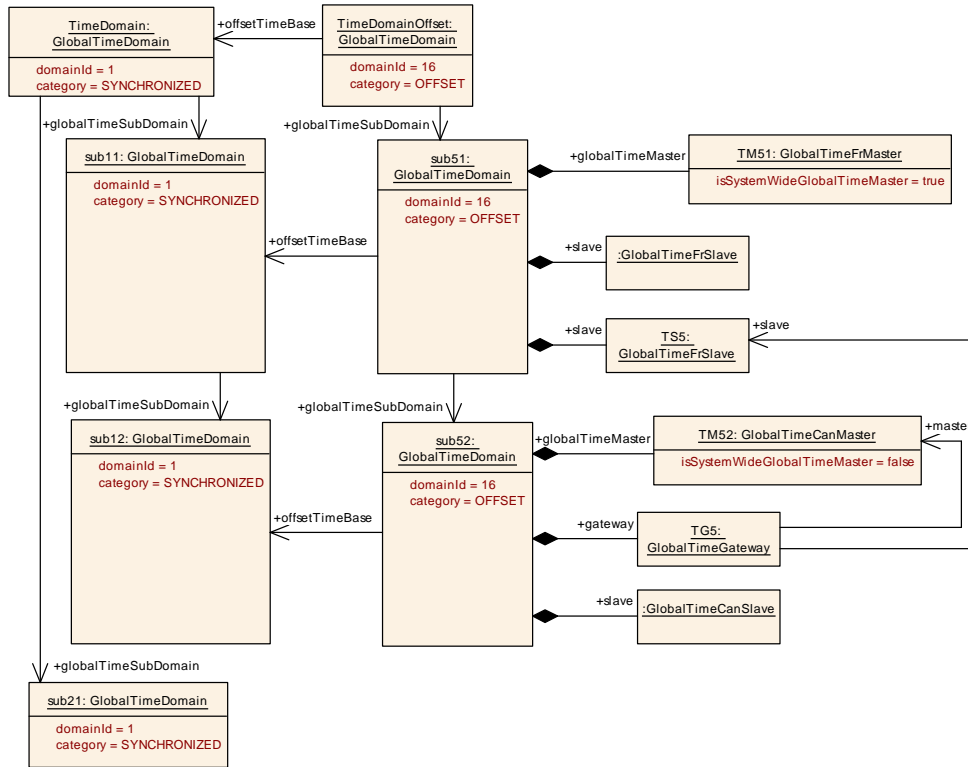


Figure 9.5: System Description of Offset Time Sync example

Class	GlobalTimeDomain			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the ability to define a global time domain. Tags: atp.recommendedPackage=GlobalTimeDomains			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
debounceTime	TimeValue	0..1	attr	Defines the minimum amount of time between two time sync messages are transmitted.
domainId	PositiveInteger	1	attr	This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.
gateway	GlobalTimeGateway	*	aggr	A GlobalTimeGateway may exist in the context of a GlobalTimeDomain to actively update the global time information as it is routed from one GlobalTimeDomain to another. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
globalTimeCorrectionProps	GlobalTimeCorrectionProps	0..1	aggr	Defintion of attributes for rate and offset correction.





Class	GlobalTimeDomain			
globalTimeDomainProperty	AbstractGlobalTimeDomainProps	0..1	aggr	Additional properties of the GlobalTimeDomain. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
globalTimeMaster	GlobalTimeMaster	0..1	aggr	This represents the single master of a GlobalTimeDomain. A GlobalTimeDomain may have no GlobalTimeDomain.master, e.g. when it gets its time from a GPS receiver. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
globalTimeSubDomain	GlobalTimeDomain	*	ref	By this means it is possible to create a hierarchy of subDomains where one global time domain can declare one or more other global time domains as its subDomains. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
offsetTimeDomain	GlobalTimeDomain	0..1	ref	Reference to a synchronized time domain this offset time domain is based on. The reference source is the offset time domain. The reference target is the synchronized time domain.
pduTriggering	PduTriggering	0..1	ref	This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
slave	GlobalTimeSlave	*	aggr	This represents the collections of slaves of the GlobalTimeDomain. A GlobalTimeDomain may have no GlobalTimeDomain.slaves, e.g. when it propagates its time directly to sub domains. Stereotypes: atpVariation Tags: vh.latestBindingTime=postBuild
syncLossTimeout	TimeValue	0..1	attr	This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain.

Table 9.1: GlobalTimeDomain

Class	AbstractGlobalTimeDomainProps (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This abstract class enables a GlobalTimeDomain to specify additional properties.			
Base	ARObject			
Subclasses	CanGlobalTimeDomainProps , EthGlobalTimeDomainProps , FrGlobalTimeDomainProps			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 9.2: AbstractGlobalTimeDomainProps

Class	GlobalTimeMaster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the generic concept of a global time master.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	GlobalTimeCanMaster , GlobalTimeEthMaster , GlobalTimeFrMaster , UserDefinedGlobalTimeMaster			
Attribute	Type	Mult.	Kind	Note
communication Connector	Communication Connector	1	ref	The GlobalTimeMaster is bound to the Communication Connector.
immediate ResumeTime	TimeValue	0..1	attr	Defines the minimum time between an "immediate" message and the next periodic message.
isSystemWide GlobalTime Master	Boolean	1	attr	If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.
syncPeriod	TimeValue	1	attr	This represents the period. Unit: seconds

Table 9.3: GlobalTimeMaster

Class	GlobalTimeSlave (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the generic concept of a global time slave.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	GlobalTimeCanSlave , GlobalTimeEthSlave , GlobalTimeFrSlave , UserDefinedGlobalTimeSlave			
Attribute	Type	Mult.	Kind	Note
communication Connector	Communication Connector	1	ref	The GlobalTimeSlave is bound to the Communication Connector.
followUp TimeoutValue	TimeValue	0..1	attr	Rx timeout for the follow-up message.
timeLeapFuture Threshold	TimeValue	0..1	attr	Defines the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value.
timeLeap HealingCounter	PositiveInteger	0..1	attr	Defines the required number of updates to the Time Base where the time difference to the previous received value has to remain within the bounds of timeLeapFuture Threshold and timeLeapPastThreshold until that Time Base is considered healed.
timeLeapPast Threshold	TimeValue	0..1	attr	Defines the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value.

Table 9.4: GlobalTimeSlave

Class	GlobalTimeGateway			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the ability to define a time gateway for establishing a global time domain over several communication clusters.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
host	EcuInstance	1	ref	The GlobalTimeGateway is hosted by the referenced Ecu Instance.





Class	GlobalTimeGateway			
master	GlobalTimeMaster	1	ref	This represents the master of the global time gateway.
slave	GlobalTimeSlave	1	ref	This represents the slave of the GlobalTimeGateway.

Table 9.5: GlobalTimeGateway

[TPS_SYST_02115] Applicability of [GlobalTimeDomain.globalTimeDomain-Property](#) [The defined properties at [GlobalTimeDomain.globalTimeDomain-Property](#) may be defined individually per [GlobalTimeDomain](#). This allows to define different value sets for each [GlobalTimeDomain](#) and any of the sub-domains.]()

[TPS_SYST_02163] Applicability of [syncLossTimeout](#) [[GlobalTimeDomain.syncLossTimeout](#) shall be specified for [GlobalTimeDomains](#) that have an aggregated slave and for all other cases this attribute is not applicable.]()

Class	GlobalTimeCorrectionProps			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This meta-class defines the attributes for rate and offset correction.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
offsetCorrectionAdaptionInterval	TimeValue	0..1	attr	Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation.
offsetCorrectionJumpThreshold	TimeValue	0..1	attr	Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump.
rateCorrectionMeasurementDuration	TimeValue	0..1	attr	Definition of the time span which is used to calculate the rate deviation.
rateCorrectionsPerMeasurementDuration	PositiveInteger	0..1	attr	Defines the number of simultaneous rate measurements to determine the current rate deviation.

Table 9.6: GlobalTimeCorrectionProps

9.3 Detailed Description of Global Time Synchronization

This chapter describes how the concept of *global time synchronization* is applied to various communication bus systems.

Although the characteristics of the supported bus systems differ widely in terms of their communication behavior, the modeling is actually quite similar for all of the supported bus systems.

9.3.1 Time Synchronization over CAN

This chapter described the detailing of how the concept of *global time synchronization* is applied to the CAN bus in particular.

The implementation of *global time synchronization* on the CAN bus is modeled by means of `GlobalTimeCanMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `GlobalTimeCanSlave`, which is derived from `GlobalTimeSlave`.

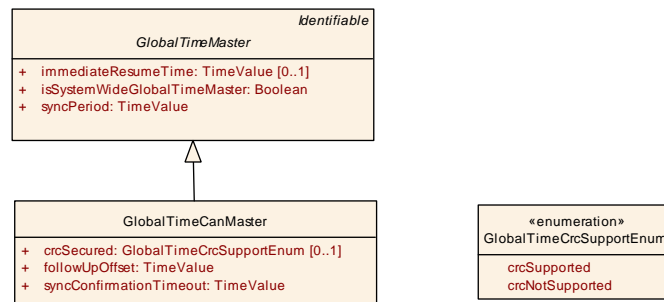


Figure 9.6: Modeling of the `GlobalTimeCanMaster`

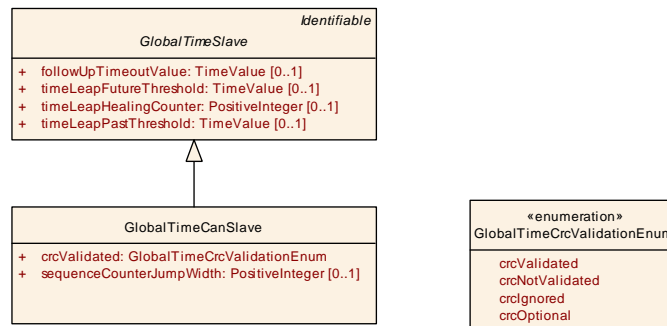


Figure 9.7: Modeling of the `GlobalTimeCanSlave`

In addition to the CAN specific Master and Slave properties CAN specific `CanGlobalTimeDomainProps` can be described.

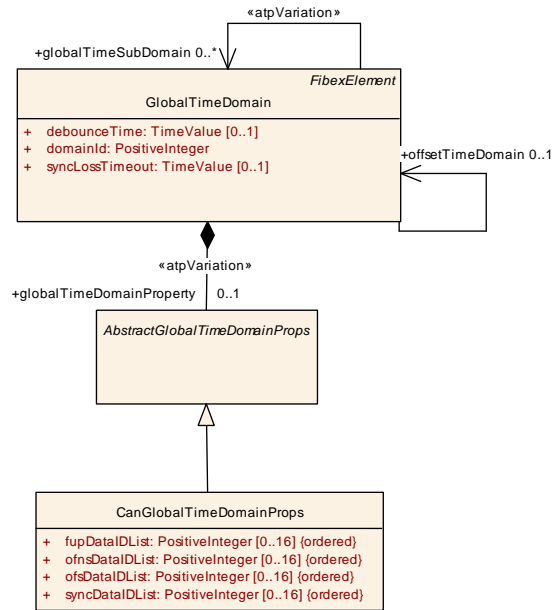


Figure 9.8: Modeling of the CAN specific `CanGlobalTimeDomainProps`

Class	GlobalTimeCanMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::CAN			
Note	This represents the specialization of the GlobalTimeMaster for the CAN communication.			
Base	ARObject, GlobalTimeMaster, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
crcSecured	GlobalTimeCrcSupport Enum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.
followUpOffset	TimeValue	1	attr	This represents the offset of the Follow-Up message with respect to the SYNC message
sync Confirmation Timeout	TimeValue	1	attr	This represents the value for the confirmation timeout. Unit: seconds.

Table 9.7: GlobalTimeCanMaster

Class	GlobalTimeCanSlave			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::CAN			
Note	This represents the specialization of the GlobalTimeSlave for the CAN communication.			
Base	ARObject, GlobalTimeSlave, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
crcValidated	GlobalTimeCrc ValidationEnum	1	attr	Definition of whether or not validation of the CRC is supported.
sequence CounterJump Width	PositiveInteger	0..1	attr	Specifies the maximum allowed gap of the sequence counter between two SYNC resp. two OFS messages.

Table 9.8: GlobalTimeCanSlave

Class	CanGlobalTimeDomainProps			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::CAN			
Note	Enables the definition of Can Global Time specific properties.			
Base	ARObject, AbstractGlobalTimeDomainProps			
Attribute	Type	Mult.	Kind	Note
fupDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for FUP messages to calculate CRC.
ofnsDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for OFNS messages to calculate CRC.
ofsDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for OFS messages to calculate CRC.
syncDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for SYNC messages to calculate CRC.

Table 9.9: CanGlobalTimeDomainProps

9.3.2 Time Synchronization over Ethernet

This chapter described the detailing of how the concept of *global time synchronization* is applied to the Ethernet bus in particular. For details concerning the functional behavior please refer to [38].

The implementation of *global time synchronization* on the Ethernet bus is modeled by means of `GlobalTimeEthMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `GlobalTimeEthSlave`, which is derived from `GlobalTimeSlave`.

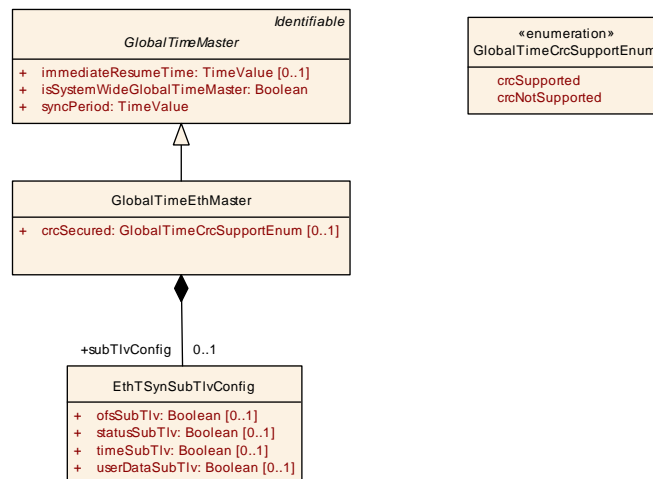


Figure 9.9: Modeling of the `GlobalTimeEthMaster`

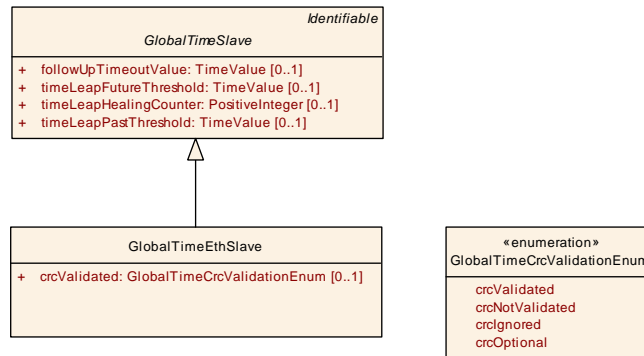


Figure 9.10: Modeling of the GlobalTimeEthSlave

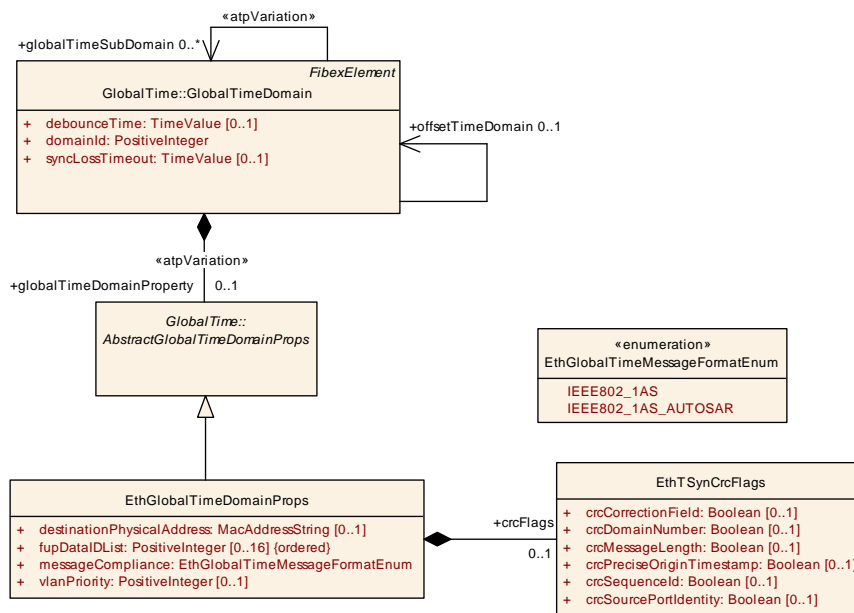


Figure 9.11: Modeling of the EthGlobalTimeDomainProps

Class	GlobalTimeEthMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	This represents the specialization of the GlobalTimeMaster for Ethernet communication.			
Base	ARObject, GlobalTimeMaster, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
crcSecured	GlobalTimeCrcSupportEnum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.
subTlvConfig	EthTSynSubTlvConfig	0..1	aggr	Defines the subTLV fields which shall be included in the time sync message.

Table 9.10: GlobalTimeEthMaster

Class	EthTSynSubTlvConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Defines the subTLV fields which shall be included in the time sync message.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
ofsSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used.
statusSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used.
timeSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used.
userDataSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used.

Table 9.11: EthTSynSubTlvConfig

Class	GlobalTimeEthSlave			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	This represents the specialization of the GlobalTimeSlave for Ethernet communication.			
Base	<i>ARObject</i> , GlobalTimeSlave , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
crcValidated	GlobalTimeCrcValidationEnum	0..1	attr	Definition of whether or not validation of the CRC is supported.

Table 9.12: GlobalTimeEthSlave

Class	EthGlobalTimeDomainProps			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Enables the definition of Ethernet Global Time specific properties.			
Base	<i>ARObject</i> , AbstractGlobalTimeDomainProps			
Attribute	Type	Mult.	Kind	Note
crcFlags	EthTSynCrcFlags	0..1	aggr	Defines the fields of the message which shall be taken into account for CRC calculation and verification.
destination Physical Address	MacAddressString	0..1	attr	Defines the MAC multicast address the Ethernet time sync messages are communicated on.
fupDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for FUP messages to calculate CRC.
managed CouplingPort	EthGlobalTimeManagedCouplingPort	*	aggr	Collection of CouplingPorts which are managed in the scope of this Ethernet GlobalTimeDomain.
message Compliance	EthGlobalTimeMessageFormatEnum	1	attr	Defines the compliance of the Ethernet time sync messages to specific standards.
vlanPriority	PositiveInteger	0..1	attr	Defines which VLAN priority shall be assigned to a time sync message in case the message is sent using a VLAN tag.

Table 9.13: EthGlobalTimeDomainProps

Class	EthTSynCrcFlags			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Defines the fields of the message which shall be taken into account for CRC calculation and verification.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
crcCorrectionField	Boolean	0..1	attr	CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.
crcDomainNumber	Boolean	0..1	attr	DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.
crcMessageLength	Boolean	0..1	attr	MessageLength from the Follow_Up Message Header shall be included in CRC calculation.
crcPreciseOriginTimestamp	Boolean	0..1	attr	PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.
crcSequenceld	Boolean	0..1	attr	Sequenceld from the Follow_Up Message Header shall be included in CRC calculation.
crcSourcePortIdentity	Boolean	0..1	attr	SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.

Table 9.14: EthTSynCrcFlags

Enumeration	EthGlobalTimeMessageFormatEnum
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH
Note	Specifies which message formats are available to for the Ethernet time sync protocol.
Literal	Description
IEEE802_1AS	Message format according to IEEE 802.1AS standard. Tags: atp.EnumerationLiteralIndex=0 xml.name=IEEE802-1AS
IEEE802_1AS_AUTOSAR	Message format according to IEEE 802.1AS standard with AUTOSAR extensions. Tags: atp.EnumerationLiteralIndex=1 xml.name=IEEE802-1AS-AUTOSAR

Table 9.15: EthGlobalTimeMessageFormatEnum

[constr_3312] Consistency of **vlanPriority** and **EthernetCommunicationConnector** [A **GlobalTimeEthMaster** refers to an **EthernetCommunicationConnector** in the role **communicationConnector**. If that **EthernetCommunicationConnector** is referenced by an **EthernetPhysicalChannel** in the role **commConnector** and the **EthernetPhysicalChannel** has a **vLan** tag defined via the **VlanConfig** then the **GlobalTimeDomain** of the **GlobalTimeEthMaster** shall aggregate **EthGlobalTimeDomainProps** in the role **globalTimeDomainProperty** and the attribute **EthGlobalTimeDomainProps.vlanPriority** shall exist.]
()

In Ethernet networks the usage of Ethernet switches introduces another layer of delay in the transportation of data. This of course also applies to the global time synchronization messages and there are means to compensate these delays available in AUTOSAR.

In order to cope with delays on global time sync of Ethernet transport technology two use-cases are supported:

- an ECU is connected to an Ethernet switch but does not manage the switch (see section 9.3.2.2)
- an ECU is connected to an Ethernet switch and also manages this switch (see section 9.3.2.3)

The `CouplingPort` is used in either use-case to describe the connection of the ECU to the Ethernet network / switch. Thus there are some attributes related to the `CouplingPort` which apply to both use-cases.

9.3.2.1 Time Synchronization and Ethernet propagation delay

The propagation delay measurement is applicable to the `CouplingPort` in scope of the global time synchronization.

The default propagation delay time (which is used if propagation delay measurement is disabled or is not yet measured) is defined at the `GlobalTimeCouplingPortProps` with the attribute `propagationDelay`. The `GlobalTimeCouplingPortProps` are aggregated at the `CouplingPortDetails` in the role `globalTimeProps`.

Whether an ECU shall initiate a propagation delay measurement at a certain `CouplingPort` and on a specific `GlobalTimeDomain` is defined by the attribute `pdelayRequestPeriod` at the `EthGlobalTimeManagedCouplingPort`.

[TPS_SYST_03016] Applicability of `EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod` [When `EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod` is not defined or has the value 0 then initiation of propagation delay measurement is disabled for the `CouplingPort` referenced by `couplingPort` and the `GlobalTimeDomain` the `EthGlobalTimeManagedCouplingPort` belongs to.]()

Whether an ECU shall respond to propagation delay measurement at a certain `CouplingPort` and on a specific `GlobalTimeDomain` is defined by the attribute `pdelayResponseEnabled` at the `EthGlobalTimeManagedCouplingPort`.

9.3.2.2 Time Synchronization and Ethernet connection

In case the ECU is directly connected to the Ethernet network and does not manage an Ethernet switch (of course there may be Ethernet switches used in the topology, but for this use-case these switches are not visible to the description of this ECU) the Ethernet time synchronization only needs to cope with the connection of this ECU to the Ethernet network. Considering the example in figure 9.13 this applies to the ECUs TS1, TS2, TS3, and TM which are just connected to the Ethernet switches but do not manage any Ethernet switch.

[TPS_SYST_03017] Reference to **CouplingPort** in the context of a **GlobalTimeDomain** [In case a **GlobalTimeDomain** is communicated via a **CouplingPort** and the respective ECU does not manage an Ethernet switch then the reference **EthGlobalTimeManagedCouplingPort.couplingPort** shall reference a **CouplingPort** which is aggregated by the **EthernetCommunicationController** in the role **couplingPort**. The **EthernetCommunicationController** itself shall be referenced by a **GlobalTimeMaster** or **GlobalTimeEthSlave** (via the **CommunicationConnector**) and that **GlobalTimeMaster** or **GlobalTimeEthSlave** shall be aggregated by the **GlobalTimeDomain** initially mentioned.]()

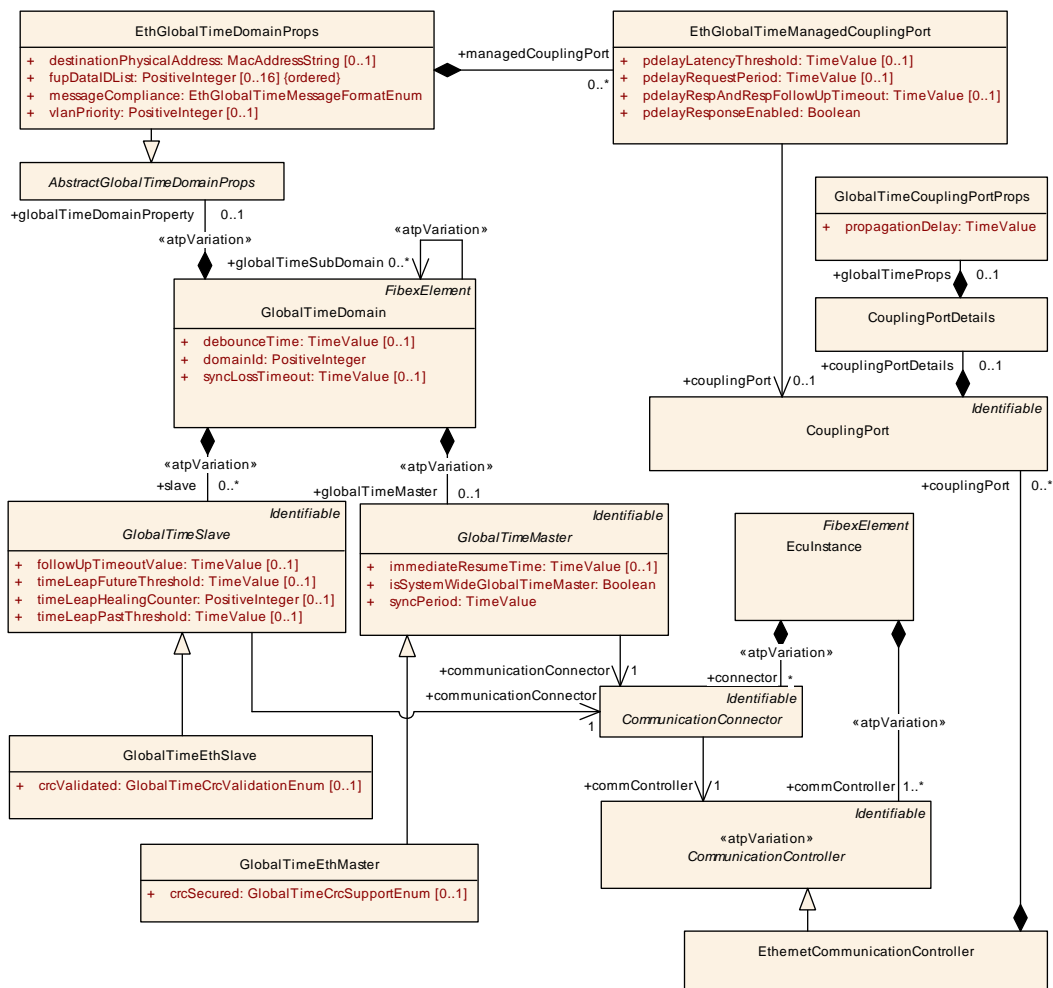


Figure 9.12: Overview of the Ethernet time sync in relation with a **CouplingPort of an ECU**

9.3.2.3 Time Synchronization and managed Ethernet switch

In case an ECU manages an Ethernet switch then that management ECU can basically be the **GlobalTimeEthMaster** or the **GlobalTimeEthSlave** located (see also figure 9.13). For the description of the time delay compensation on System Template level this does not matter.

It is essential to configure all possible time synchronization communication paths between the involved entities.

In case of ECU A in figure 9.13 the `GlobalTimeEthMaster` shall

- refer to the `CouplingPort 0` since this is where
 - all time sync messages will be sent out by the `GlobalTimeEthMaster`
 - all adjusted follow up messages will be sent out by the `GlobalTimeEthMaster`
- refer to the `CouplingPort 1` because this is where the Ethernet switch is connected to the management ECU A
- refer to the `CouplingPorts 2, 3, and 4` since this is where the time sync messages will be forwarded by the switch
- not refer to the `CouplingPort 5` because this one is not involved in that `GlobalTimeDomains` communication

In case of ECU B in figure 9.13 the `GlobalTimeEthSlave` shall

- refer to the `CouplingPort 0` since this is where
 - the time sync messages will be received by the `GlobalTimeEthSlave`
 - all adjusted follow up messages will be sent out by the `GlobalTimeEthSlave`
- refer to the `CouplingPort 1` because this is where the Ethernet switch is connected to the management ECU B
- refer to the `CouplingPort 3` because this is where the time sync messages will be received from the `GlobalTimeEthMaster` on ECU TM
- refer to the `CouplingPorts 2 and 4` since this is where the time sync messages will be forwarded by the switch
- not refer to the `CouplingPort 5` because this one is not involved in that `GlobalTimeDomains` communication

Please note that the non-involvement of the `CouplingPort 5` is used for illustration purposes. It would also be possible to involve `CouplingPort 5` in that `GlobalTimeDomain` definition although currently there is no ECU as an `GlobalTimeEthSlave` defined. In that case the `CouplingPort 5` is prepared to be connected to an ECU with an `GlobalTimeEthSlave` later.

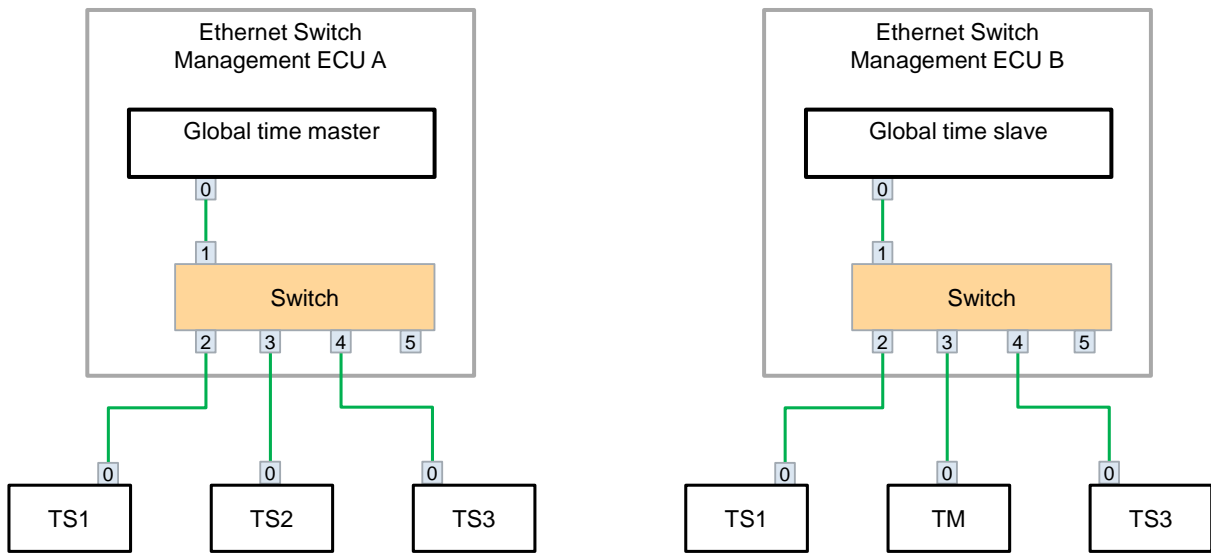


Figure 9.13: Example of a managed Ethernet Switch

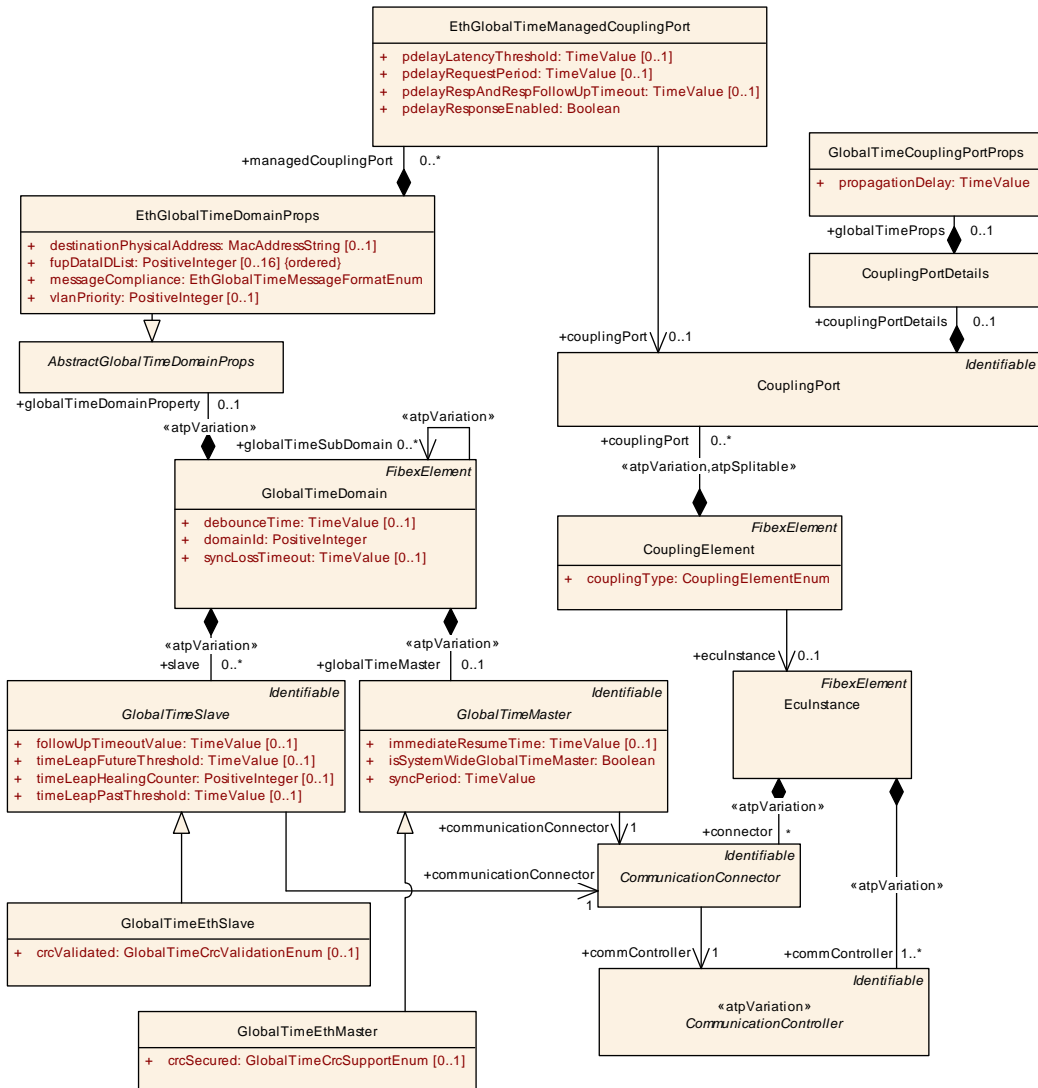


Figure 9.14: Overview of the Ethernet time sync in relation with a **CouplingPort** of an Ethernet switch

Class	EthGlobalTimeManagedCouplingPort			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Specifies a CouplingPort which is managed by an Ethernet Global Time Domain.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
couplingPort	CouplingPort	0..1	ref	Defines which CouplingPort is managed by this EthGlobal TimeManagedCouplingPort.
pdelayLatency Threshold	TimeValue	0..1	attr	Threshold for calculated Pdelay. If a measured Pdelay exceeds pdelayLatencyThreshold, the measured Pdelay value is discarded.
pdelayRequest Period	TimeValue	0..1	attr	Defines the period for the pdelay request messages.





Class		EthGlobalTimeManagedCouplingPort		
pdelayRespAndRespFollowUpTimeout	TimeValue	0..1	attr	Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 or not defining this attribute deactivates this timeout observation.
pdelayResponseEnabled	Boolean	1	attr	Defines whether PDELAY RESPONSE and PDELAY RESPONSE FOLLOW UP shall be sent on this Coupling Port.

Table 9.16: EthGlobalTimeManagedCouplingPort

Class		GlobalTimeCouplingPortProps		
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines properties for the usage of the CouplingPort in the scope of Global Time Sync.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
propagationDelay	TimeValue	1	attr	If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter defines a fixed value for the propagation delay.

Table 9.17: GlobalTimeCouplingPortProps

9.3.3 Time Synchronization over FlexRay

This chapter described the detailing of how the concept of *global time synchronization* is applied to the Flexray bus in particular.

The implementation of *global time synchronization* on the Flexray bus is modeled by means of `GlobalTimeFrMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `GlobalTimeFrSlave`, which is derived from `GlobalTimeSlave`.

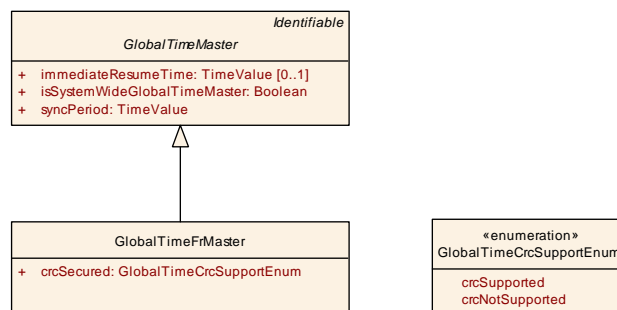


Figure 9.15: Modeling of the `GlobalTimeFrMaster`

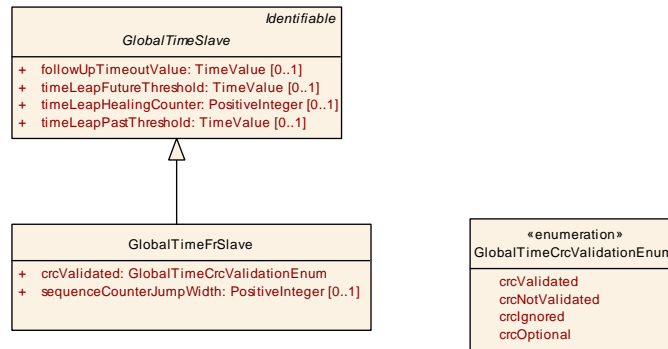


Figure 9.16: Modeling of the GlobalTimeFrSlave

In addition to the FlexRay specific Master and Slave properties FlexRay specific `FrGlobalTimeDomainProps` can be described.

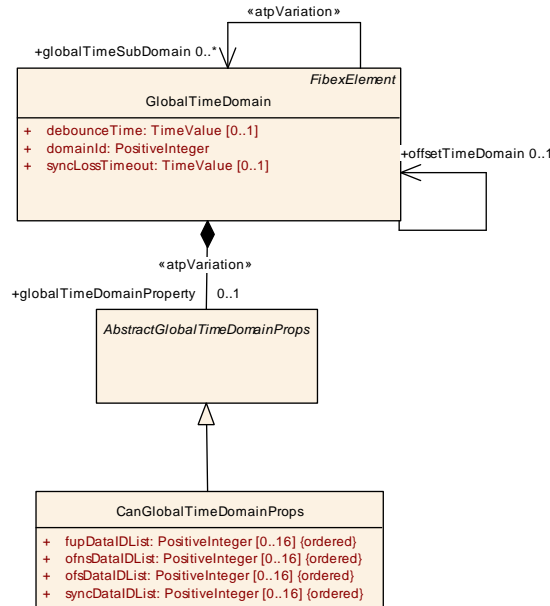


Figure 9.17: Modeling of the FlexRay specific FrGlobalTimeDomainProps

Class	GlobalTimeFrMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::FR			
Note	This represents the specialization of the GlobalTimeMaster for Flexray communication.			
Base	ARObject, GlobalTimeMaster, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
crcSecured	GlobalTimeCrcSupport Enum	1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.

Table 9.18: GlobalTimeFrMaster

Class	GlobalTimeFrSlave			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::FR			
Note	This represents the specialization of the GlobalTimeSlave for Flexray communication.			
Base	ARObject, GlobalTimeSlave , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
crcValidated	GlobalTimeCrcValidationEnum	1	attr	Definition of whether or not validation of the CRC is supported.
sequenceCounterJumpWidth	PositiveInteger	0..1	attr	Specifies the maximum allowed gap of the sequence counter between two SYNC resp. two OFS messages.

Table 9.19: GlobalTimeFrSlave

Class	FrGlobalTimeDomainProps			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::FR			
Note	Enables the definition of Flexray GlobalTime specific properties.			
Base	ARObject, AbstractGlobalTimeDomainProps			
Attribute	Type	Mult.	Kind	Note
ofsDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for OFS messages to calculate CRC.
syncDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for SYNC messages to calculate CRC.

Table 9.20: FrGlobalTimeDomainProps

9.3.4 Time Synchronization by user defined Timebase Provider

This chapter describes the details of how the concept of global time synchronization is applied to user defined Timebase Providers. The implementation of global time synchronization by user defined timebase providers is modeled by means of [UserDefinedGlobalTimeMaster](#), a concrete subclass of [GlobalTimeMaster](#). A similar approach applies for the [UserDefinedGlobalTimeSlave](#), which is derived from [GlobalTimeSlave](#).

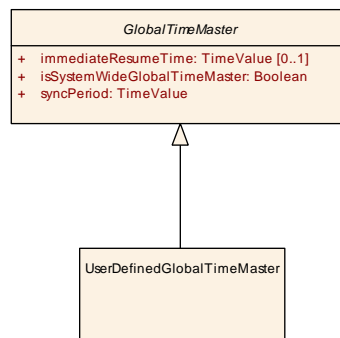


Figure 9.18: Modeling of the [UserDefinedGlobalTimeMaster](#)

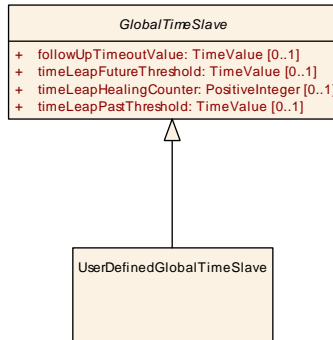


Figure 9.19: Modeling of the `UserDefinedGlobalTimeSlave`

Class	UserDefinedGlobalTimeMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::UserDefined			
Note	This represents the specialization of the GlobalTimeMaster for user defined communication.			
Base	ARObject, GlobalTimeMaster, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 9.21: UserDefinedGlobalTimeMaster

Class	UserDefinedGlobalTimeSlave			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::UserDefined			
Note	This represents the specialization of the GlobalTimeSlave for user defined communication.			
Base	ARObject, GlobalTimeSlave, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 9.22: UserDefinedGlobalTimeSlave

9.3.5 Time Synchronization Common Properties

The purpose of this chapter is basically to provide the class tables of meta-classes taken to implement configuration properties in the context of *global time synchronization*. The specifics about how these meta-classes are used is explained in the bus-specific chapters (i.e. chapters 9.3.1, 9.3.2, and 9.3.3).

Enumeration	GlobalTimeCrcSupportEnum
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime
Note	This enumeration is used to define whether and how CRC on the TX side shall be utilized.
Literal	Description
crcNotSupported	This indicates that CRC is not supported Tags: atp.EnumerationLiteralIndex=0





<i>Enumeration</i>	GlobalTimeCrcSupportEnum
crcSupported	This indicates that CRC is supported Tags: atp.EnumerationLiteralIndex=1

Table 9.23: GlobalTimeCrcSupportEnum

<i>Enumeration</i>	GlobalTimeCrcValidationEnum
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime
Note	This enumeration provides values for the evaluation of the CRC
Literal	Description
crclgnored	The CRC is supposed to be ignored Tags: atp.EnumerationLiteralIndex=0
crcNotValidated	The CRC is not supposed to be present. If CRC is present the message is ignored. Tags: atp.EnumerationLiteralIndex=1
crcOptional	Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done. Tags: atp.EnumerationLiteralIndex=3
crcValidated	This CRC is supposed to be validated. Tags: atp.EnumerationLiteralIndex=2

Table 9.24: GlobalTimeCrcValidationEnum

10 Usage of the System Template

As introduced in [TPS_SYST_01003] the System Template is used to describe a `System` with `category` `SYSTEM_CONSTRAINT_DESCRIPTION`, a `System` with `category` `ABSTRACT_SYSTEM_DESCRIPTION` and a `System` with `category` `SYSTEM_DESCRIPTION`. `System` with `category` `SYSTEM_EXTRACT` is described in more detail in chapter 11. `System` with `category` `ECU_EXTRACT` is described in more detail in chapter 12.

Certain elements of the System Template may have a different meaning at the different stages of the AUTOSAR Methodology. The following sections describe the differences.

10.1 System Constraint Description

Meta-classes, Chapters	Usage to describe the System Constraints	Usage to describe the System Configuration
CommunicationCluster , EcuInstance (chapter 3)	<p>The Topology is completely described in the System Constraint Description.</p>	<p>The Topology description will be unchanged copied to the System Configuration description. The Topology may only be changed during another iteration development step of the whole system.</p>
FrameTriggering , PduTriggering , ISignalTriggering (chapter 6)	<p>The System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> describes all FrameTriggerings that are predefined on all CommunicationClusters of a vehicle. The predefinition of the communication matrix forces the system generator to use the given FrameTriggerings. Constraints for the system generator arise here e.g. from the used bus bandwidth, used identifiers as well as from the timing and at which position in a Frame a Pdu is transmitted on a PhysicalChannel on a CommunicationCluster. Such a manual definition of the communication can be made for any reason where it is necessary to restrict the system generator. One example is the usage of legacy EcuInstances in an AUTOSAR System. The FrameTriggerings that are transmitted or received by these legacy EcuInstances are constraints for the system generator because they cannot be changed, if the compatibility is supposed to be achieved without any changes at the legacy EcuInstances.</p>	<p>In contrary to the System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> the final System with category <code>SYSTEM_DESCRIPTION</code> contains all FrameTriggerings, PduTriggerings, ISignalTriggerings that will be sent by any EcuInstance in the car. No matter if they were predefined (system constraint) or if they were generated by the system generator. The available information, in addition to the information, which is inserted by the AUTOSAR Ecu configuration generator step, will be used as input to configure the Basic SW for the communication.</p>
Gateway (chapter 8)	<p>The System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> describes all Gateways in the system including their IPduMappings and ISignalMappings that are predefined. The reasons for such predefinitions are quite the same as for the predefinitions of the FrameTriggerings.</p>	<p>In contrary to the System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> the final System with category <code>SYSTEM_DESCRIPTION</code> describes all Gateways with all their IPduMappings and ISignalMappings. No matter if they were predefined (System Constraint) or if they were generated by the System Generator.</p>

<p>SwcToEcuMapping (subsection 5.1.1)</p>	<p>The mapping of Software Components to EcuInstances may be predefined. The predefinition will force the system generator to use the specified mapping. Thus, with the SwcToEcuMapping element it is possible to describe that one or more Software Components shall be mapped to a specific EcuInstance.</p>	<p>In a complete System with category <code>SYSTEM_DESCRIPTION</code>, all Software Components are mapped to EcuInstances.</p>
<p>MappingConstraint (subsection 5.1.4)</p> <p>ComponentClustering (subsubsection 5.1.4.1)</p> <p>ComponentSeparation (subsubsection 5.1.4.2)</p>	<p>There may be system constraints that limit the system generators freedom to map Software Components to arbitrary EcuInstances. These system constraints can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator.</p>	<p>After the mapping has been completed, the System with category <code>SYSTEM_DESCRIPTION</code> will contain mapping descriptions for all elements, and the mapping constraints are obsolete. But that does not mean that mapping constraints have to be deleted after the system generation step. By deleting the mapping constraints you would lose the information why a mapping of a Software Component to an EcuInstance is chosen.</p>
<p>DataMapping (section 5.2)</p> <p>SenderReceiverToSignalMapping (subsubsection 5.2.1.1)</p> <p>SenderReceiverToSignalGroupMapping (subsubsection 5.2.1.2)</p> <p>ClientServerToSignalMapping (subsubsection 5.2.1.3)</p>	<p>The System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> may describe the predefined mapping of Software Components to certain EcuInstances (see chapter 5.1.1). Only if such a mapping exists, it is reasonable to define the DataMapping of the data exchanged between the Software Components.</p>	<p>In contrary to the System with category <code>SYS- TEM_CONSTRAINT_DESCRIPTION</code> the final System with category <code>SYSTEM_DESCRIPTION</code> shall contain all DataMapping definitions. No matter if they were predefined (system constraint) or if they were generated by the System-Generator.</p>

<p>SignalPathConstraint (subsection 5.2.2)</p> <p>CommonSignalPath (subsubsection 5.2.2.1)</p> <p>ForbiddenSignalPath (subsubsection 5.2.2.2)</p> <p>PermissibleSignalPath (subsubsection 5.2.2.3)</p> <p>SeparateSignalPath (subsubsection 5.2.2.4)</p>	<p>It can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator, which specific way a VariableDataPrototype or ClientServerOperation should take in the network without defining in which Pdu and Frame it is transmitted.</p>	<p>SignalPathConstraints are not an obligatory part of the System with category SYSTEM_DESCRIPTION. In the final System with category SYSTEM_DESCRIPTION every ISignal is assigned to a Pdu and every Pdu is assigned to a Frame. Thereby the paths of VariableDataPrototypes or ClientServerOperations on the network are implicitly described. But that does not mean that the SignalPathConstraints have to be deleted after the system generation step. By deleting the SignalPathConstraints you would lose the information why you have chosen e.g. a specific mapping of an ISignal into a Pdu. If you extend or change the system at a later stage the missing SignalPathConstraints could lead to not wanted signal mappings by the System Generator.</p>
--	---	--

Table 10.1: Usage of the System Template

10.2 Abstract System Description

[TPS_SYST_01134] Abstract System Description [Due to the fact that the functional view on vehicle system can differ from the actual technical definition of the software-architectures of individual EcuInstances the System Template optionally allows to define a System with category ABSTRACT_SYSTEM_DESCRIPTION.]()

[TPS_SYST_01135] Refactoring of an Abstract System Description into a project specific technical view of the software architecture [The System with category ABSTRACT_SYSTEM_DESCRIPTION concentrates on the functional aspects of the system design and provides an own abstract VFB. During the further activities this abstract view shall be refactored into a more project specific technical view of the software architecture.

It is important to note that during the refactoring of the System with category ABSTRACT_SYSTEM_DESCRIPTION into the System with category SYSTEM_DESCRIPTION no restrictions to the allowed actions apply (This is in contrast to the activity of deriving the System with category SYSTEM_EXTRACT from the System with category SYSTEM_DESCRIPTION, see [TPS_SYST_01123].)()

[TPS_SYST_01136] ViewMapSet and ViewMap are used to trace the transformations between different models [The ViewMapSet and ViewMap elements are used

to trace the transformations between different models within the AUTOSAR environment.]()

These classes are described in more detail in the Generic Structure Template [2].

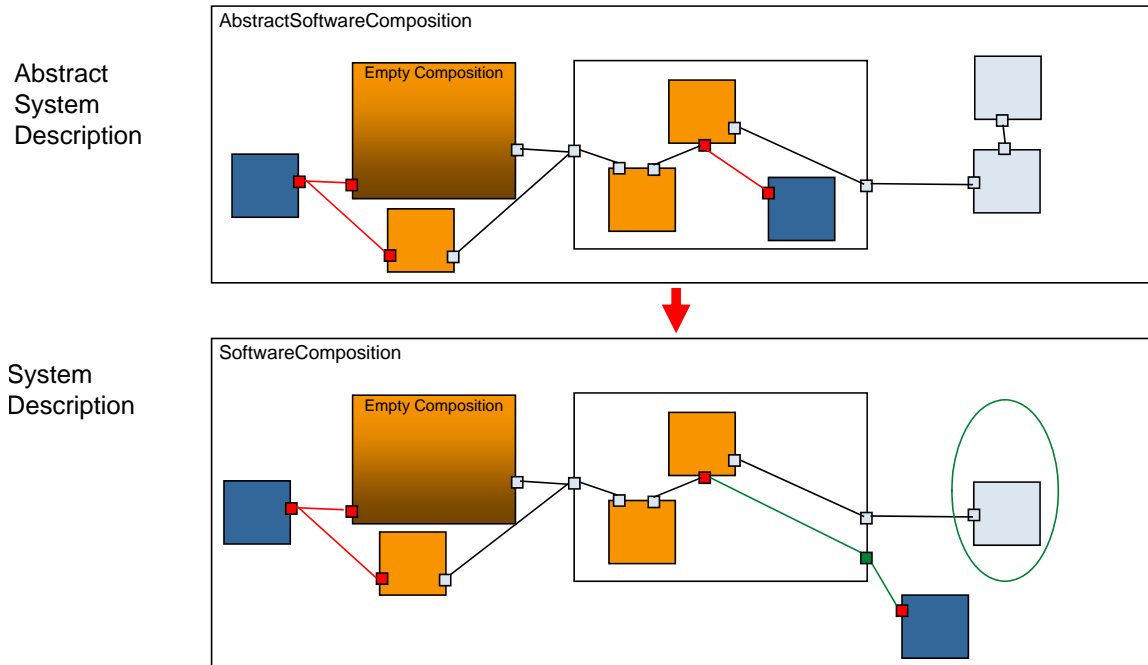


Figure 10.1: Abstract System Description refactoring to a System Description

11 System Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR work product `System` with `category` `SYSTEM_EXTRACT`, based on Meta Model elements contained in the System Template and Software Component Template.

The `System` with `category` `SYSTEM_EXTRACT` is introduced to allow a collaboration between an OEM and a Supplier.¹ The OEM/Supplier Collaboration scenario is described in more detail in chapter 11.1.

The OEM is often only interested in the required functionality and the integration of the functionality into the `System`. Thus the OEM provides a basis for designing a subsystem, which is developed by the supplier. One difference to the `System` with `category` `ECU_EXTRACT` is that the `System` with `category` `SYSTEM_EXTRACT` is not fully decomposed and still needs to be refined before it forms the basis for the ECU configuration. Another difference is that a `System` with `category` `SYSTEM_EXTRACT` is not fixed to an `EcuInstance`.

[TPS_SYST_01123] System Extract may cover one or many `EcuInstances` [The `System` with `category` `SYSTEM_EXTRACT` may cover one or many `EcuInstances`.]
()

The `System` with `category` `SYSTEM_EXTRACT` is using the same meta model elements as the `System` with `category` `SYSTEM_DESCRIPTION`. The `System` with `category` `SYSTEM_DESCRIPTION` is a special case of a `System` with `category` `SYSTEM_EXTRACT`. From the technical point of view there is no difference. The distinction is only made for the sake of Methodology [4].

In the `System` with `category` `SYSTEM_EXTRACT` the OEM strips all information from the `System` with `category` `SYSTEM_DESCRIPTION` that is not needed for the definition of the subsystem. There is one exception to this simple "remove" rule: the communication mapping may need to be extended, which will be described in more detail in chapter 11.2.

[TPS_SYST_03000] Co-existing `System` with `category` `SYSTEM_DESCRIPTION` and `System` with `category` `SYSTEM_EXTRACT` [In order to be able to handle one `System` with `category` `SYSTEM_DESCRIPTION` and one or several `Systems` with `category` `SYSTEM_EXTRACT` within the same workspace it shall be possible to provide different full qualified names to the elements of `System` with `category` `SYSTEM_EXTRACT`.] (*RS_SYST_00045*)

When different `Systems` with various categories co-exist it is possible to define `ViewMap` and `ViewMapSet` between their elements according to [TPS_SYST_01136].

¹ Collaboration scenarios between different departments of an OEM are also supported by the `System` with `category` `SYSTEM_EXTRACT`. For the sake of simplicity such scenarios are not addressed here.

In contrast to the *System* with *category* ECU_EXTRACT the *System* with *category* SYSTEM_EXTRACT may contain *CompositionSwComponentTypes*. Empty *CompositionSwComponentTypes* in the *System* with *category* SYSTEM_EXTRACT represent subsystems that need to be refined by a Supplier. Figure 11.1 shows an example where a *System* with *category* SYSTEM_DESCRIPTION is stripped down to a subsystem.

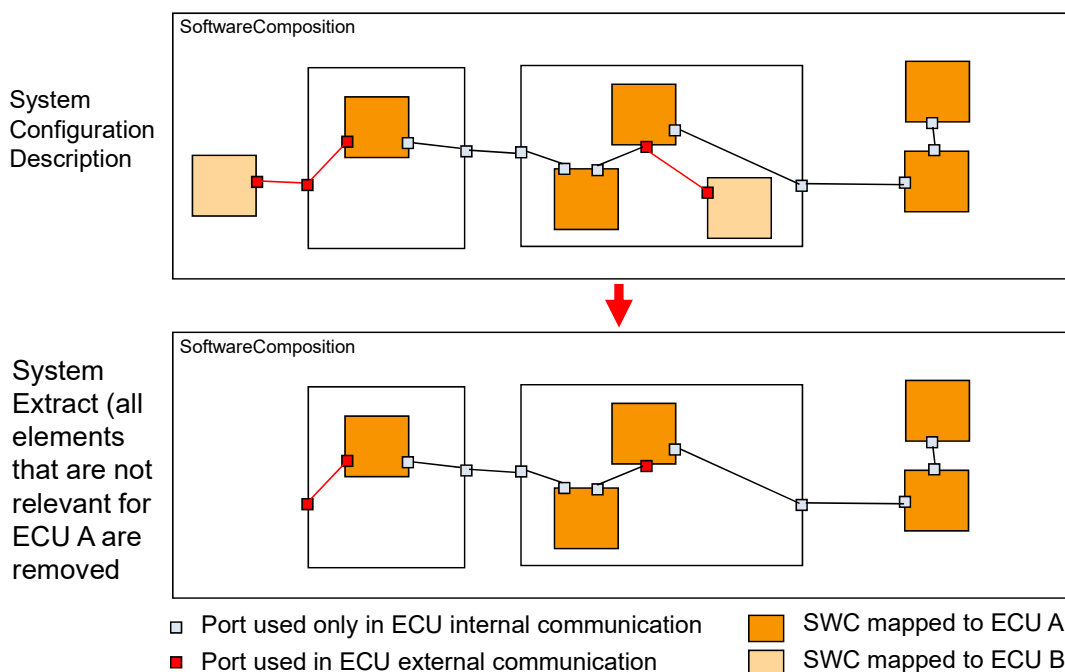


Figure 11.1: System Extract creation: irrelevant elements are removed from the System Description

11.1 OEM/Supplier Collaboration Scenario

In an important collaboration scenario, an OEM commissions a supplier to provide implementations of one or more functionalities to be integrated into an AUTOSAR system in the form of Application Components. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System VFB rather than the internal structure of such a component. On the other hand, the supplier, delivering both the component implementation in combination with the ECU it is destined to run on, may claim the internal structure of such a higher-level component contains substantial intellectual property, and hence may not want to disclose its internal works to the OEM.

Effectively, the use case can be described in the following manner:

- The OEM generates a *System* with *category* SYSTEM_EXTRACT from the *System* with *category* SYSTEM_DESCRIPTION. From the *System* with *category* SYSTEM_DESCRIPTION all elements are removed that are not relevant for the design of the subsystem, such as SW components or topology elements.

- The OEM can deliver a sub-structure of Software Compositions or even Atomic Software Components in the `System` with `category` `SYSTEM_EXTRACT`. But the `System` with `category` `SYSTEM_EXTRACT` can also contain empty Software Compositions. The OEM shall have the possibility to define only the outer shell of a Software Composition that is to implement a certain functionality. Such an empty `CompositionSwComponentType` does contain all the provided and required ports with the included `ReceiverComSpecs` and `SenderComSpecs` describing the requested component's outside communication needs. But it does not need to contain `SwComponentPrototypes` or `SwConnectors` at this stage.
- Such empty components are added to a System's VFB, the outside ports are connected with other components in the VFB. However, at this stage the inner structure of such `CompositionSwComponentType` can still be left empty.
- The `System` with `category` `SYSTEM_EXTRACT` contains the mapping of components to the target `EcuInstances`, including the empty compositions. Signal mappings affecting the empty compositions are targeting the `CompositionSwComponentType`'s ports.
- The OEM delivers the `System` with `category` `SYSTEM_EXTRACT` to the Supplier.
- The Supplier adds the substructure to the empty `CompositionSwComponentTypes` by adding `SwComponentPrototypes` and `SwConnectors`. This once more leads to a hierarchical VFB, effectively the Supplier creates a local System Description for his subsystem.
- The Supplier adjusts the Signal mappings to the actual ports of the inner `AtomicSwComponentType` prototype.
- The Supplier generates the `System` with `category` `ECU_EXTRACT` from his ECU-local system description. The resulting `System` with `category` `ECU_EXTRACT` does not include prototypes of type `CompositionSwComponentType` any longer.
- Based on this `System` with `category` `ECU_EXTRACT` the actual ECU configuration is done.

When the supplier receives the `System` with `category` `SYSTEM_EXTRACT` from the OEM he has basically two choices how to proceed:

1. The Supplier takes the `System` with `category` `SYSTEM_EXTRACT` of the OEM as the structural basis for the ECU development. In this case the following steps may follow:
 - The Supplier adds the substructure to the empty `CompositionSwComponentTypes` by adding `SwComponentPrototypes` and `SwConnectors`. This once more leads to a hierarchical VFB, effectively the Supplier creates a local System Description for his subsystem (`System` with `category` `ECU_SYSTEM_DESCRIPTION`).

- The Supplier adjusts the Signal mappings to the actual ports of the inner `AtomicSwComponentType` prototype.
2. The Supplier creates an own structure to base the ECU development on `System` with `category` `ECU_SYSTEM_DESCRIPTION` and perform a view mapping between the OEM's `System` with `category` `SYSTEM_EXTRACT` and the `System` with `category` `ECU_SYSTEM_DESCRIPTION`. In this case the following steps may follow:
 - The Supplier develops an own structure how the ECU shall be designed but needs to respect the required outer boundary of the OEM's required communication behavior (`ReceiverComSpecs` and `SenderComSpecs`).
 - The Supplier adjusts the Signal mappings to the actual ports of the inner `AtomicSwComponentType` prototype.

When the design of the `System` with `category` `ECU_SYSTEM_DESCRIPTION` is complete the following steps follow:

- The Supplier generates the `System` with `category` `ECU_EXTRACT` from his `System` with `category` `ECU_SYSTEM_DESCRIPTION`. The resulting `System` with `category` `ECU_EXTRACT` does not include prototypes of type `CompositionSwComponentType` any longer.
- Based on this `System` with `category` `ECU_EXTRACT` the actual ECU configuration is done.

11.2 Data Mapping in the System Extract

As mentioned before, there is a slight complication to the simple "remove" rule. This can be shown best with an example.

Example: Assume a simple topology with two `EcuInstances` A and B and three `Pdus` X (sent from A to B), Y (sent from B to A) and Z (sent from B to A) as shown in Figure 11.2.

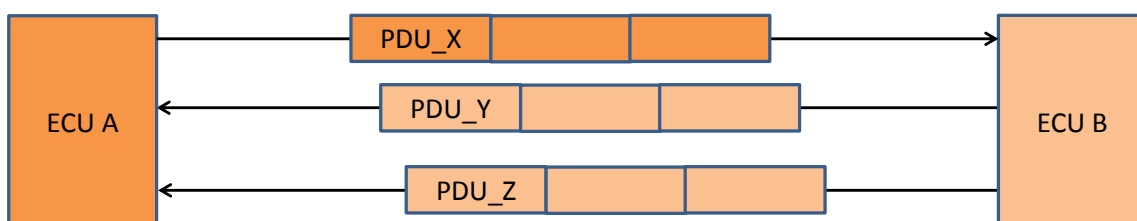
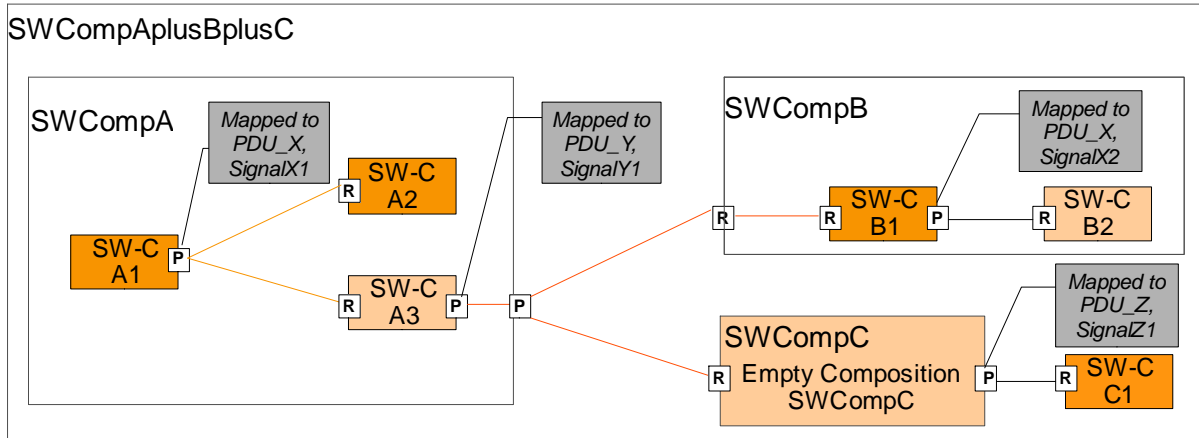


Figure 11.2: Example topology with two `EcuInstances` and three `Pdus` exchanged between them

Furthermore assume a composition of software-components realized by the meta-class `CompositionSwComponentType` as shown in Figure 11.3. It consists of six `SwComponentPrototypes` 'A1' to 'A3' (aggregated in composition 'SwCompA'),

'B1' / 'B2' (aggregated in composition 'SWCompB'), 'C1' (aggregated in composition 'SWCompAplusBplusC') and an empty composition 'SWCompC'.

The overall composition 'SWCompAplusBplusC' aggregates 'SwCompA', 'SWCompB', the empty 'SWCompC' and the *SwComponentPrototype* 'C1'.



- SW Component mapped to ECU A
- SW Component mapped to ECU B

Figure 11.3: Example SW composition with mapping information

The atomic *SwComponentPrototypes* 'A1', 'A2', 'B1' and 'C1' are mapped to 'ECU A'. The atomic *SwComponentPrototypes* 'A3', 'B2' and the empty composition 'SWCompC' are mapped to 'ECU B'. The data sent from

- 'A1' to 'A3' is mapped to 'PDU_X', 'SignalX1',
- 'B1' to 'B2' is mapped to 'PDU_X', 'SignalX2' and
- 'A3' to 'B1' and 'A3' to 'SWCompC' is mapped to 'PDU_Y', 'SignalY1'
- 'SWCompC' to 'C1' is mapped to 'PDU_Z', 'SignalZ1'

As usual, the data mapping rules refer to the *VariableDataPrototype* in the *PPortPrototype* of the sending SW component. Note that *DataMappings* can be performed on compositions and on atomic *SwComponentPrototypes* as described in chapter 5.2.1. ²

Figure 11.4 shows how the System extract for ECU A and for ECU B of this SW composition would look like: Only those elements are included that are relevant for the subsystem.

²Data mapping is allowed on empty compositions and on compositions that contain atomic *SwComponentPrototypes*.

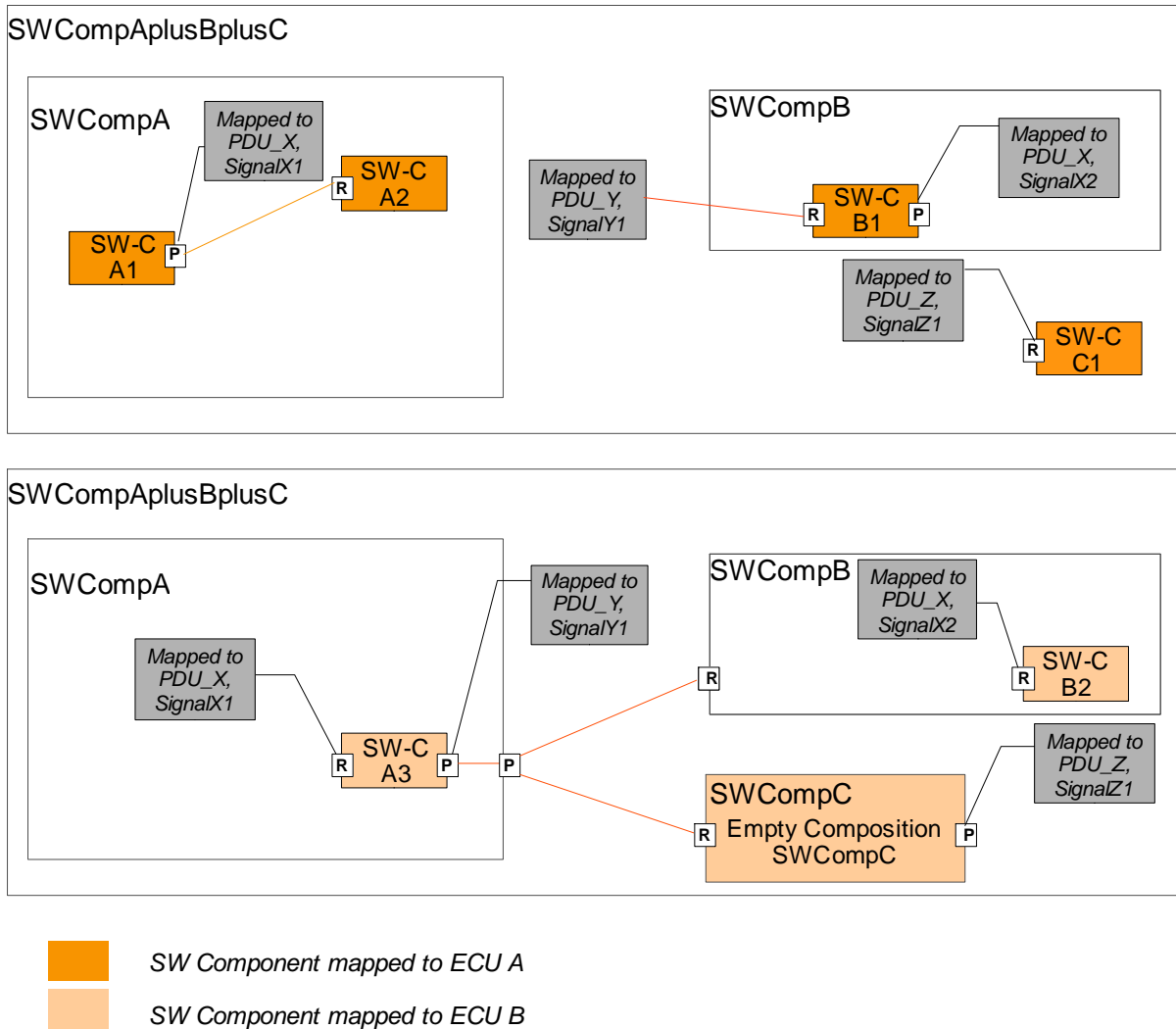


Figure 11.4: Example System extract for ECU A (upper figure) and ECU B (lower figure) of above introduced composition

In both figures all `SwComponentPrototypes` and compositions that are mapped onto the `EcuInstance` are included. The `SwConnector` between these `SwComponentPrototypes` are also included. Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

`SwConnectors` that were used to connect to SW components that are not included in the System Extract are not included. Instead, the mapping to an `ISignal` in a `Pdu` is used to identify the source/destination of that data.

The problem that new mapping rules have to be added arises for example in the System Extract for 'ECU A' with the mapping to 'PDU Y', 'SignalY1': Since SW component 'A3', which was referenced in the original mapping, is no longer included, the data mapping needs a new data element in a port to reference to. In the example, it is the required port of 'B1', so that the Supplier has the information that B1 receives the data via 'PDU Y'.

11.3 SW component inclusion and top level data mapping

In section 11.2 the approach is to provide the `DataMapping` on the `PortPrototypes` of the `SwComponentPrototypes` which are mapped to one `EcuInstance`. Since the granularity of mapping `SwComponentPrototypes` to `EcuInstances` is possible for individual atomic `SwComponentPrototypes` this approach may result in many `DataMappings` from different software component `PortPrototypes` to the same `SystemSignal` (depending where in the hierarchical structure they are located).

An alternative approach is to provide the complete communication information of the whole System Extract on the `RootSwCompositionPrototype` and perform the `DataMapping` on the `PortPrototypes` of the `RootSwCompositionPrototype` only. This approach is illustrated in figure 11.5.

`PortPrototypes` are created on the `RootSwCompositionPrototype` representing the external communication of this `EcuInstance`. `DelegationSwConnectors` are created to establish the communication of the external software components with the software components inside the local `EcuInstance`.

In figure 11.5 the software components X, Y and Z are mapped to remote `EcuInstances`. Their communication needs are collected in `PortPrototypes` on the `RootSwCompositionPrototype` and the communication is delegated via `SwConnectors` inside the hierarchical software component structure.

In this example the approach for X and Y is trivial since there are only some `DelegationSwConnectors` required to connect the `PortPrototypes` of the `RootSwCompositionPrototype` with the `PortPrototypes` of the respective `SwComponentPrototypes`.

But for `SwComponentPrototype` Z the approach needs to be extended, because the communication on system level is designed to happen inside the composition V. In this case the communication needs to be delegated out of the composition (creation of `DelegationSwConnectors` inside the composition V) to be visible in the `RootSwCompositionPrototype`. Then again the approach of connection to the `RootSwCompositionPrototype` can be applied.

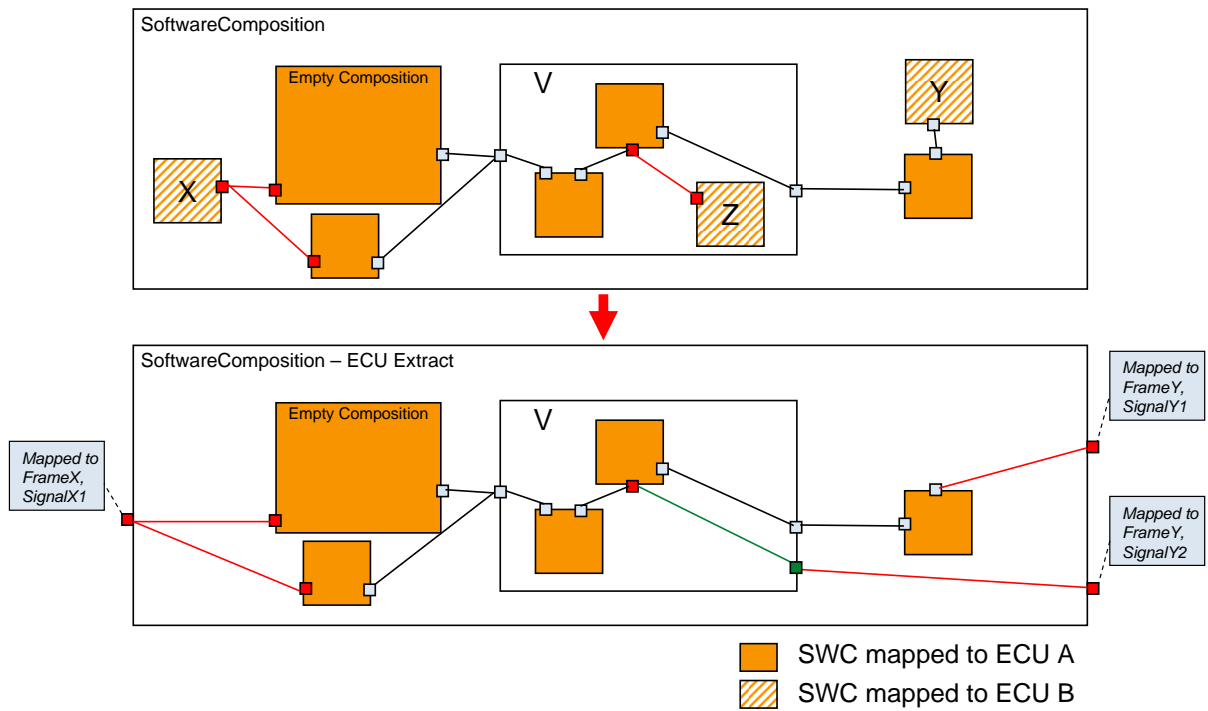


Figure 11.5: Example with software components mapped to two ECUs

11.4 Port-Interface Mapping in the System Extract

A similar problem as the one with `DataMappings` described in chapter 11.2 and chapter 11.3 exists for the `PortInterfaceMappings` as well. To illustrate this Figure 11.6 depicts an example with software components mapped to two different ECUs.

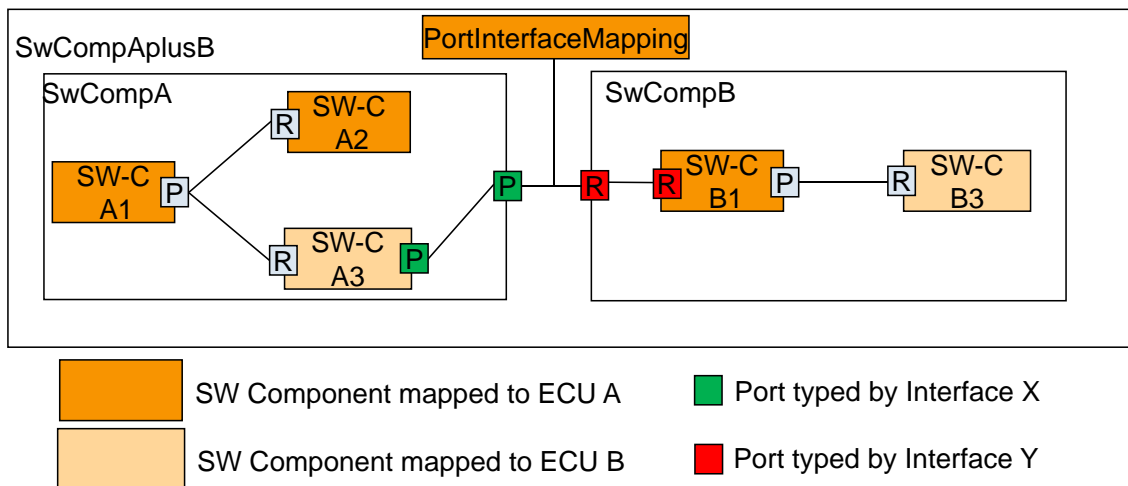


Figure 11.6: Example with software components mapped to two ECUs

Hereby the `PPortPrototype` typed with `PortInterface X` of `SWCompA` is connected with the `RPortPrototype` typed with `PortInterface Y` of `SWCompB` by means of an `AssemblySwConnector`. This `AssemblySwConnector` has an attached `PortInterfaceMapping` to perform a mapping between the elements (see chapter 4.3.1.5 of [5]) of the two otherwise incompatible `PortInterfaces X` and `Y`.

A System Extract for ECU A is now created by applying the approach described in chapter 11.3, i.e., by providing the complete communication information of the whole System Extract on the `RootSwCompositionPrototype` and performing the `DataMapping` on the `PortPrototypes` of the `RootSwCompositionPrototype` only.

When doing this however the following two additional things have to be considered:

- The `PortInterfaceMapping` shall be preserved during this process
- The information about the `PortInterfaces` referenced by the `PortPrototypes` connected by the `AssemblySwConnector` referencing the `PortInterfaceMapping` shall be preserved during this process

Just as in the approach described in chapter 11.3 `PortPrototypes` are created on the `RootSwCompositionPrototype` representing the external communication of this `EcuInstance`. The `RPortPrototypes` however are not typed by the `PortInterface X` of the `RPortPrototypes` of the `SwComponentPrototypes` inside `ECU A` (`SWCompB` in the example) but by the `PortInterface Y` of the `PPortPrototype` which was connected to the `RPortPrototypes` by means of the `AssemblySwConnector`. Afterwards the just like in the approach described in chapter 11.3

`DelegationSwConnectors` are created to connect the `PortPrototypes` of the `RootSwCompositionPrototype` with the corresponding `RPortPrototypes` of the `SwComponentPrototypes` inside *ECU A*.

This however yields a `DelegationSwConnector` between `RPortPrototype` typed by `PortInterface Y` (which has been created on the `RootSwCompositionPrototype`) and the `RPortPrototype` typed by `PortInterface X` of *SWCompB*. In order to perform a mapping between the elements of these otherwise incompatible interfaces, the `PortInterfaceMapping` which has initially been referred to by the `AssemblySwConnector` needs to be referenced by the `DelegationSwConnector`.

The final result of this process is depicted in Figure 11.7

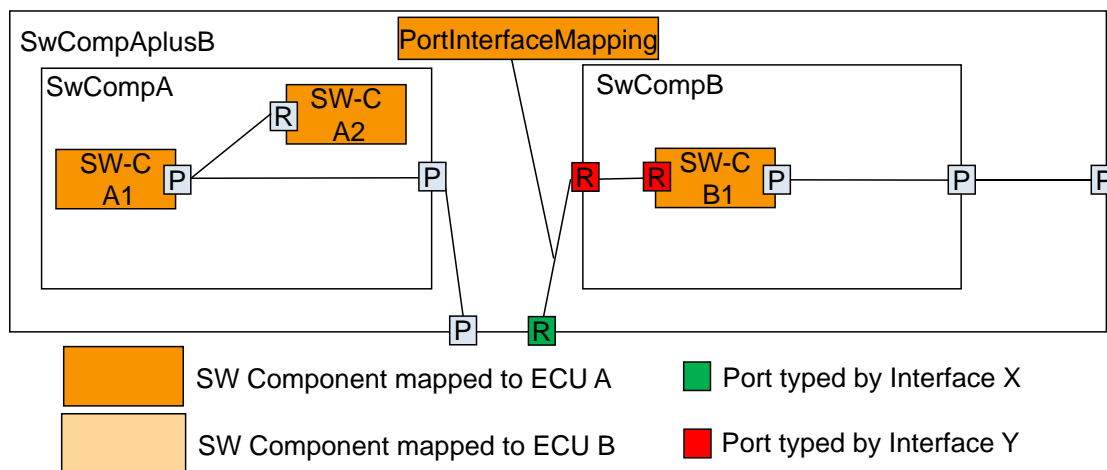


Figure 11.7: Example with software components mapped to two ECUs

12 ECU Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR `System` with `category` `ECU_EXTRACT`, based on Meta Model elements contained in the System Template and Software Component Template.

The `System` with `category` `ECU_EXTRACT` represents the view of one specific `EcuInstance` onto the overall `System` with `category` `SYSTEM_DESCRIPTION`. The `System` with `category` `ECU_EXTRACT` forms the basis for configuring that particular `EcuInstance` in focus.

For instance, RTE configuration fundamentally depends on the number and types of `SwComponentPrototypes` deployed onto the `EcuInstance`; Services are configured according to those Software Components' `ServiceNeeds`; the COM-stack BSW

modules will be configured considering the `EcuInstance`'s participation in the overall System Network Topology and Communication.

[TPS_SYST_01139] Ecu Extract covers exactly one `EcuInstance` [The `System` with `category` `ECU_EXTRACT` shall only contain the subset of information derived from the `System` with `category` `SYSTEM_DESCRIPTION` or `System` with `category` `SYSTEM_EXTRACT` relevant for configuring the targeted `EcuInstance`.]()

In order to keep ECU configuration focused and manageable despite the complexity of a full System Configuration, all other information shall be stripped from the `System` with `category` `SYSTEM_DESCRIPTION` or from the `System` with `category` `SYSTEM_EXTRACT` when creating the `System` with `category` `ECU_EXTRACT`.

AUTOSAR VFB Descriptions naturally form hierarchies of `CompositionSwComponentTypes`. Consequently, in the System Configuration the SWC-related information for different `EcuInstances` is not separated but in general is intermingled. In contrast, for the task of ECU configuration (RTE configuration, Service Configuration, Measurement and Calibration) a hierarchically “flat view” on the `SwComponentPrototypes` running on the `EcuInstances` is preferable over a hierarchical view, which is more favored by application-software development. Thus, deriving an `System` with `category` `ECU_EXTRACT` actually is a model transformation, following a set of rules described in the following sections.

As System- and ECU development typically happens in iterations, the use case of repeatedly extracting the information from an incrementally changing System Configuration needs to be considered. In particular, it shall be possible to detect changes between consecutively generated ECU extracts in order to selectively update the existing ECU configuration (12.5).

AUTOSAR supports the definition and consequently the handling of Variability in the System Configuration. According to the specified binding time associated with a particular `VariationPoint`, typically some of these variants will already be resolved at the time of a `System` with `category` `ECU_EXTRACT`. If however the binding time occurs in a later stage of the AUTOSAR methodology, i.e. during ECU Configuration or later, the variability needs to be carried over to the `System` with `category` `ECU_EXTRACT`. This also holds true for Variation points that ultimately are resolved at system configuration time but affect post-build configuration parameters. (12.6)

The `System` with `category` `ECU_EXTRACT` logically forms one entity. Therefore, for ease of readability the rest of the chapter assumes just one file, “the XML file”. However, it explicitly is allowed to split the `System` with `category` `ECU_EXTRACT` over several files.

12.1 Topology

Only those Topology elements relevant for the `EcuInstance` in scope are taken over from the `System` with `category` `SYSTEM_DESCRIPTION` into the `System` with `category` `ECU_EXTRACT`.

- The `System` with `category` `ECU_EXTRACT` is always associated with exactly one `EcuInstance`. Therefore exactly one `EcuInstance` is included along with all classes included in `EcuInstance` by composition: `CommunicationControllers` and `CommunicationConnectors` with all their `CommConnectorPorts`.
- A `CommunicationCluster` is included along with all its `PhysicalChannels` if at least one `PhysicalChannel` is used by the `EcuInstance`. In other words, if at least one of the included `CommunicationConnectors` is referenced by any of a `CommunicationCluster`'s `PhysicalChannels`, the whole `CommunicationCluster` and all its `PhysicalChannels` are included.
- From the used `PhysicalChannels`, only those `FrameTriggerings`, `PduTriggerings`, `ISignalTriggerings` shall be included that are used by the `EcuInstance`, e.g. they are associated with a `FramePort`, `IPduPort`, `ISignalPort` belonging to one of the `EcuInstance`'s `CommunicationConnectors`. *Note:* Including just a subset of a `PhysicalChannel`'s `FrameTriggerings`, `PduTriggerings`, `ISignalTriggerings` is possible without changing the `PhysicalChannel` itself because of the `<<splittable>>` stereotype applied on the `PhysicalChannel / FrameTriggering, PduTriggering, ISignalTriggering` composition.

As the Topology elements are not modified when taken over into the `System` with `category` `ECU_EXTRACT`, their package structure and short names are not touched (see section 12.4.1).

12.2 Top-level Software Composition

In the `System` with `category` `SYSTEM_DESCRIPTION` the application software composition is hierarchic by nature as described in chapter 4. When mapping `SwComponentPrototypes` onto concrete `EcuInstances` using the `SwcToEcuMapping` class (section 5.1.1), either `SwComponentPrototypes` of type `AtomicSwComponentType`, or `SwComponentPrototypes` of type `CompositionSwComponentType` are deployed onto one specified `EcuInstance`.

In order to obtain this ECU-centric view, the hierarchical structure of the `System` with `category` `SYSTEM_DESCRIPTION` needs to be transformed into a 1-layer representation, where one distinguished `CompositionSwComponentType` hosts all `SwComponentPrototypes` of type `AtomicSwComponentType` to run on the `EcuInstance`. In the `System` with `category` `ECU_EXTRACT` the resulting `RootSwCompositionPrototype` is a flat structure where the included `SwComponentPrototypes` become

real SWC instances, reflecting the actual resource needs on the targeted [EcuInstance](#).

[TPS_SYST_01140] Ecu Extract contains only [SwComponentPrototypes](#) of type [AtomicSwComponentType](#) in the [RootSwCompositionPrototype](#) [The [System](#) with [category](#) ECU_EXTRACT only contains [SwComponentPrototypes](#) of type [AtomicSwComponentType](#) in the [RootSwCompositionPrototype](#) which are effectively mapped onto the [EcuInstance](#) in focus.]()

The transformation from hierarchical to flat Software Component structure includes a number of steps, to be performed per ECU. The list below outlining this process assumes that the extraction is done for the first time; if an [System](#) with [category](#) ECU_EXTRACT already exists from a previous development cycle, the extract shall merely be updated instead of created; for more details on iterative development see section 12.5.

- Create the one [CompositionSwComponentType](#) which will represent the ECU's SW subsystem (in further steps referred to as ECU flat view)
- To this ECU flat view, add a [SwComponentPrototype](#) for each instance of any [AtomicSwComponentType](#) mapped onto the [EcuInstance](#). Copy all the identifiable information from the originating [SwComponentPrototype](#), but assign a unique short name to the new element. The newly created [SwComponentPrototypes](#) are typed by the original [AtomicSwComponentType](#).
- Unroll the connector paths leading to and from the included components:
 - For ECU internal communication, use [AssemblySwConnector](#) to connect [PortPrototypes](#).
 - For ECU external communication, add delegated [PortPrototypes](#) to the ECU flat view [CompositionSwComponentType](#). The delegated [PPortPrototypes](#) shall be typed with the [PortInterface](#) of the corresponding [PPortPrototype](#) of the included [SwComponentPrototypes](#) that are used for the external communication. If the original [AssemblySwConnector](#) references a [PortInterfaceMapping](#) then the delegated [RPortPrototypes](#) shall be typed with the [PortInterface](#) of the [PPortPrototype](#) initially connected (possibly by a chain of multiple original [DelegationSwConnectors](#) and the one original [AssemblySwConnector](#) referencing the [PortInterfaceMapping](#)) to this [PPortPrototype](#) of the included [SwComponentPrototypes](#) that are used for the external communication. Each delegated [PortPrototype](#) shall be connected via a [DelegationSwConnector](#) with [PortPrototypes](#) of the included [SwComponentPrototypes](#) that are used for the external communication.
 - [VariableDataPrototypes](#) and [ClientServerOperations](#) of the delegated [PortPrototypes](#) are mapped to [SystemSignals](#).
- If the [System](#) with [category](#) SYSTEM_DESCRIPTION prescribes an [Implementation](#) for a [SwComponentPrototype](#) by using [SwcToImplMapping](#), a

corresponding constraint needs to be created in the `System` with `category ECU_EXTRACT` of the targeted `EcuInstance`. The `SwcToImplMapping`'s `component` reference needs to be adjusted to the flat representation, while maintaining the original reference to the `Implementation`.

- Only `ComSpecs` on the `PortPrototypes` of atomic software components are relevant for the RTE generator (see [TPS_SWCT_01568]). The existence of `ComSpecs` on composition level can be taken into account for setting the values on the atomic level as the atomic level gets created. On the other hand there is no obligation to respect the `ComSpec` settings on the composition level for the creation of the atomic level. Finally the approach for the creation of `ComSpec` values on atomic level depends on OEM preferences.

Figure 12.1 illustrates the process of flattening the hierarchical Software Composition into an ECU Flat View representation, as outlined in the previous paragraphs. The following sections explain the concrete transformation steps in more detail.

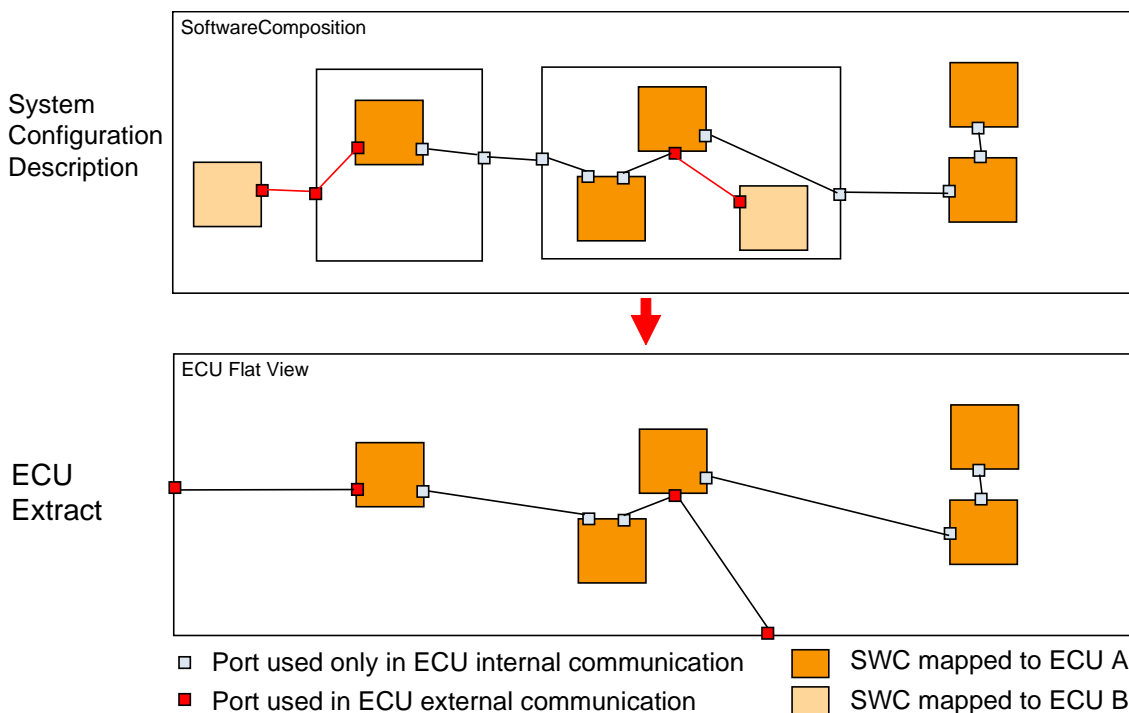


Figure 12.1: Flattening of a hierarchic Software Composition into an ECU Flat View, and the distinction between ports used in internal and those used in external communication.

Please note that instantiation specific scheduling of runnables shall be maintained when generating a `System` with `category ECU_EXTRACT`. This maintenance covers the rewrite of the `instanceRef` to the `RTEEvent` respectively the aggregation of the `instantiationRTEEventProps` to the next `CompositionSwComponentType`.

12.2.1 ECU Flat view

The first step of extracting the ECU specific Software View is the creation of a new `CompositionSwComponentType` (further referred to as ECU flat view). This new element serves as a container for collecting all `SwComponentPrototypes` of type `AtomicSwComponentType` deployed on the `EcuInstance`. In order to include the ECU flat view into the actual `System` with `category` `ECU_EXTRACT`, the `System` shall have its child class `RootSwCompositionPrototype` pointing to this ECU flat view.

Next, all `SwcToEcuMappings` present in the `System` with `category` `SYSTEM_DESCRIPTION` need to be analyzed according to the precedence rules (Section 5.1.1) in order to establish the exact set of `AtomicSwComponentType` instances to be included on this `EcuInstance`.

For each of these component instances, regardless of their order of depth in the System Configuration Description's Component hierarchy, exactly one `SwComponentPrototype` shall be created in the ECU flat view `CompositionSwComponentType`. The new element's description and type information shall be taken over from the original `SwComponentPrototype` as present in the `System` with `category` `SYSTEM_DESCRIPTION`. As an important exception to this rule, the `SwComponentPrototype`'s `shortName` shall be unique in the name space formed by the ECU flat view.

The special case of prototypes of `ParameterSwComponentTypes` and `ServiceProxySwComponentTypes` is treated in almost the same way. The only difference is that these component types can be instantiated at most once per `EcuInstance` and that for a given prototype in the `System`, instances on several `EcuInstances` can be created. The replication of `ParameterSwComponentTypes` and `ServiceProxySwComponentTypes` on several `EcuInstances` does not require any special treatment of their communication properties. For `ParameterSwComponentTypes` there are `SwConnectors` defined but no communication is involved. For more details see [TPS_SWCT_01422] in the `SwComponentTemplate` [5].

12.2.2 Internal Communication

When flattening the `RootSwCompositionPrototype` for the `System` with `category` `ECU_EXTRACT`, not only all of the ECU's Software Components are to be collected in the ECU flat view, but also any connection existing between `PortPrototypes` of the included `SwComponentPrototypes` needs to be projected onto the same `RootSwCompositionPrototype`.

In the hierarchical `RootSwCompositionPrototype`, communication between Software Components is specified by a combination of `AssemblySwConnectors` and `DelegationSwConnectors`. Several `DelegationSwConnectors` may be combined in case of a multiple-level delegation, however there will always be exactly one `AssemblySwConnector` on the outermost `CompositionSwComponentType` the port is delegated to.

In the ECU flat view, any such number of stringed together `SwConnectors` effectively connecting two `PortPrototypes` of `SwComponentPrototypes` mapped to the same `EcuInstance` are resolved to exactly one `AssemblySwConnector` per connected port pair. As there are no additional levels of “inner `SwComponentPrototypes`”. `DelegationSwConnectors` are only used to display the outside communication of an ECU in the ECU flat view.

[constr_3019] In the flat ECU extract each required interface shall be satisfied by connected provided interfaces [In case of the flat `System` with `category` `ECU_EXTRACT` all `VariableDataPrototypes` specified by the `SenderReceiverInterface` of the `RPortPrototype` need to be supplied by some of the `PPortPrototypes` being connected with `SwConnectors`.]()

For the `System` with `category` `SYSTEM_DESCRIPTION`, the Software Component Template Specification [5] allows a `CompositionSwComponentType`'s outer `PortPrototype` to be connected to more than one inner port, observing a set of compatibility rules between the outer and the inner port's `SenderReceiverInterfaces`. Such a “merge” and “split” functionality for mixing `VariableDataPrototypes` is used to limit the number of `SwConnectors` required to connect `PortPrototypes` on higher VFB levels and thus reduce complexity in the wiring of such higher-level `CompositionSwComponentTypes`. On the other hand this means that an `AssemblySwConnector` in a hierarchical VFB may expand to more than one Port-Port pair. Naturally, in the ECU flat view such “hidden” additional connections need to be made explicit by unrolling them into concrete `AssemblySwConnectors`.

Additionally `PassThroughSwConnector` may be used to map `PortInterface` elements between require and provide outer ports of `CompositionSwComponentTypes` in order to use RTE features for mapping or conversion instead of real software components. The following paragraph suggests a way how such an unrolling of `SwConnectors` may be accomplished.

Starting with the top-level `RootSwCompositionPrototype` indicating the outermost `CompositionSwComponentType`, the hierarchical software model of `SwComponentPrototypes` is recursively iterated; for each prototype of `CompositionSwComponentType`, all its `AssemblySwConnectors` are being iterated. For each such found `AssemblySwConnector` both connector ends are evaluated for `DelegationSwConnectors` further delegating the connection: In order to consider the use cases of signal “merge” and “split”, all possible communication partners need to be identified, recursively following `DelegationSwConnectors` in both directions. For each identified pair of `PPortPrototypes` and `RPortPrototypes` *actually exchanging* Information one `AssemblySwConnector` will be created in the ECU flat view.

In case that a string of `SwConnectors` started by `AssemblySwConnector` connects - directly or via `DelegationSwConnectors` - to a `PassThroughSwConnector` the `SwConnector` string is conjunct with the `SwConnector` string of the other end of the `PassThroughSwConnector`. Please note that the “merge” and “split” capability

of [DelegationSwConnectors](#) and [PassThroughSwConnectors](#) requires an individual treatment of the single [PortInterface](#) elements for the evaluation of the [SwConnector](#) string.

The following rules shall be followed when [PortInterfaceMappings](#) are converted for the flat view. [PortInterfaceMappings](#) supports the connection of Ports typed by two different [PortInterfaces](#) with unequal named [PortInterface](#) elements. More details can be found in [5].

- When unrolling a string of [SwConnectors](#) into a single [SwConnector](#) all compatibility rules and [PortInterfaceMappings](#) of the individual [SwConnector](#) need to be considered for determining which [VariableDataPrototypes](#) are being transferred between provider and requester. If [VariableDataPrototypes](#) are to be filtered out a [PortInterfaceMapping](#) shall be provided to the flatten connector such that only the transferred [VariableDataPrototypes](#) are included in the mapping.
- When unrolling a string of [SwConnectors](#) into a single [SwConnector](#) all of the [PortInterfaceMappings](#) of the individual [SwConnectors](#) need to be considered for combining them into a single [PortInterfaceMapping](#) to be associated with a new [SwConnector](#).

12.2.3 External Communication

In a [System](#) with [category](#) SYSTEM_DESCRIPTION , whenever two [SwComponentPrototypes](#) are specified to communicate across [EcuInstances](#), the details of this communication need to be fully specified: [VariableDataPrototypes](#) of [SenderReceiverInterfaces](#) and [ClientServerOperations](#) of [ClientServerInterfaces](#) are mapped onto [SystemSignals](#) as carriers of information transported across the network. According to 5.2, each instance of a [AutosarDataPrototype](#) that is to be sent over AUTOSAR COM shall be mapped exactly once onto its individual [SystemSignal](#), regardless of how many components receive the information or over how many [PhysicalChannels](#) the [SystemSignal](#) is transported.

As described above, deriving the [System](#) with [category](#) ECU_EXTRACT from [System](#) with [category](#) SYSTEM_DESCRIPTION or from [System](#) with [category](#) SYSTEM_EXTRACT means that all [SwComponentPrototypes](#) to be included in the Ecu extract are recreated in an ECU flat view. Consequently, each [DataMapping](#) concerning a [SwComponentPrototype](#) to be mapped onto the [EcuInstance](#) requires that a corresponding [DataMapping](#) be created in the [System](#) with [category](#) ECU_EXTRACT.

The ECU flat view contains delegated [PortPrototypes](#) to display the outside communication of an [EcuInstance](#). [VariableDataPrototypes](#) and [ClientServerOperations](#) of these delegated [PortPrototypes](#) are mapped to [SystemSignals](#). The original instance references indicating the mapped [AutosarDataPrototype](#) need to be adjusted to the new “flat” location in the ECU flat view.

While for the `System` with `category` `SYSTEM_DESCRIPTION` it is sufficient to describe `DataMappings` only on the provider side, the `System` with `category` `ECU_EXTRACT` additionally requires such `DataMappings` on the requiring side's ports. In this case, a new `DataMapping` maps to the existing `SystemSignal`, previously defined in the `System` with `category` `SYSTEM_DESCRIPTION` on the provider side. This is explained in more detail in figure 12.6, that is a continuation of the example from figure 11.3 in chapter 11.2.

To derive an `ECU_EXTRACT` from a `System` with `category` `SYSTEM_DESCRIPTION` or `SYSTEM_EXTRACT` unambiguous `ClientServerToSignalMappings` are required for inter-ECU n:1 client-server communication. In particular the communication path from the server to each client shall be uniquely mapped.

In this context, "communication path" encompasses the set of delegation/assembly connectors that connect the server (provide-port on SWC) through to the client (require-port on SWC).

[constr_3264] Server side `ClientServerToSignalMappings` in case of a n:1 inter-ECU client-server communication [If within the `System` with `category` `SYSTEM_DESCRIPTION` or `SYSTEM_EXTRACT` the `ClientServerToSignalMappings` for inter-ECU n:1 client-server communication are placed on the provider (server) side, then each of these `ClientServerToSignalMappings` shall (in the hierarchy of `SwComponentPrototypes`) refer to a "unique communication path" w.r.t. the `EcuInstances` the client `SwComponentPrototypes` are mapped to.]()

Note: A "unique communication path" has the property that, starting from the `ClientServerOperation` of a `PortPrototype`, a sequence of `DelegationSwConnectors` and `AssemblySwConnectors` leads to the client side and terminates at either at most one `PortPrototype` that is owned by the `AtomicSwComponentType` of the client's `SwComponentPrototype` or, if the path terminates at more than one `PortPrototype`, then the following shall hold: The clients' `SwComponentPrototypes` typed by `AtomicSwComponentTypes` owning these `PortPrototypes` shall be mapped to the same `EcuInstance` and the client identifier is used to distinguish the different clients (see [TPS_SYST_01087]).

The following example scenarios will show at which `PortPrototypes` the `ClientServerToSignalMappings` are allowed to be specified in a `System` with `category` `SYSTEM_DESCRIPTION` or `SYSTEM_EXTRACT` to derive an `ECU_EXTRACT`.

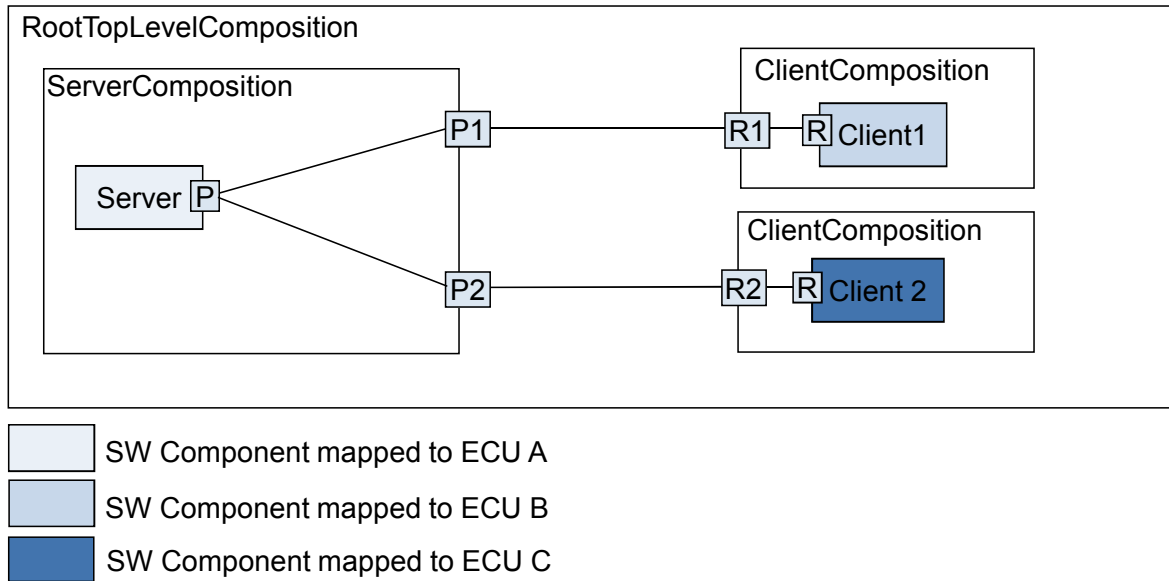


Figure 12.2: Client Server Scenario 1

For the scenario described in figure 12.2 the following statements apply:

- [ClientServerToSignalMappings](#) for the provide-port Server.P are ambiguous and thus [\[constr_3243\]](#) exists to forbid this situation.
- [ClientServerToSignalMappings](#) are permitted for ClientComposition.R1/ClientComponsonition.R2 and Client1.R/Client2.R (client-side) or for ServerComposition.P1/ServerComposition.P2 (provider-side) since there is no ambiguity.

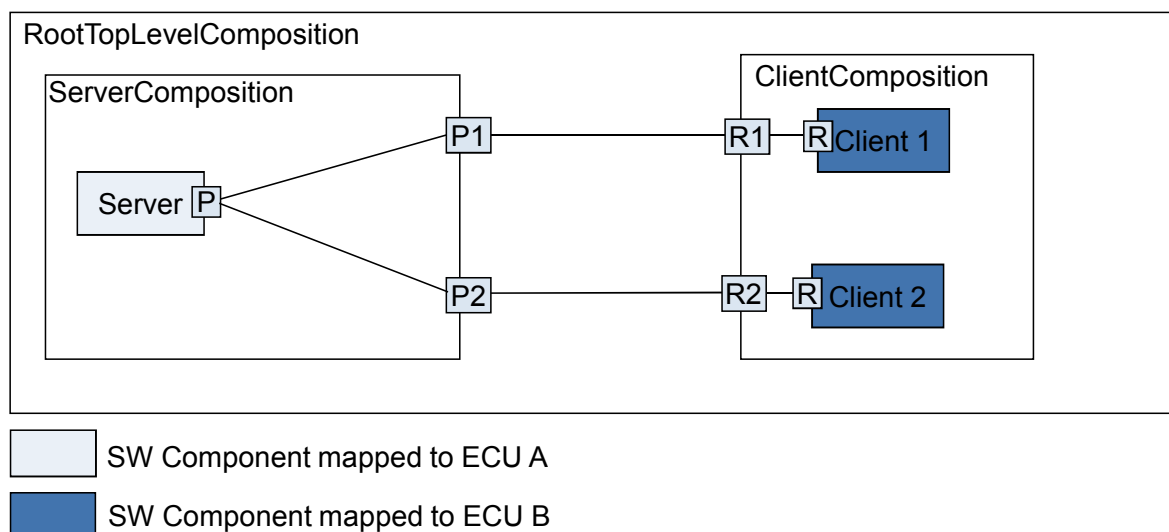


Figure 12.3: Client Server Scenario 2

For the scenario described in figure 12.3 the following statements apply:

- [ClientServerToSignalMappings](#) for the provide-ports Server.P are not ambiguous (since although there is fork in the communication path, both sub-paths end up at the same ECU).
- [ClientServerToSignalMappings](#) are permitted for ClientComposition.R1/ClientComposition.R2 and Client1.R/Client2.R (client-side) or for ServerComposition.P1/ServerComposition.P2 (provider-side) since there is no ambiguity.

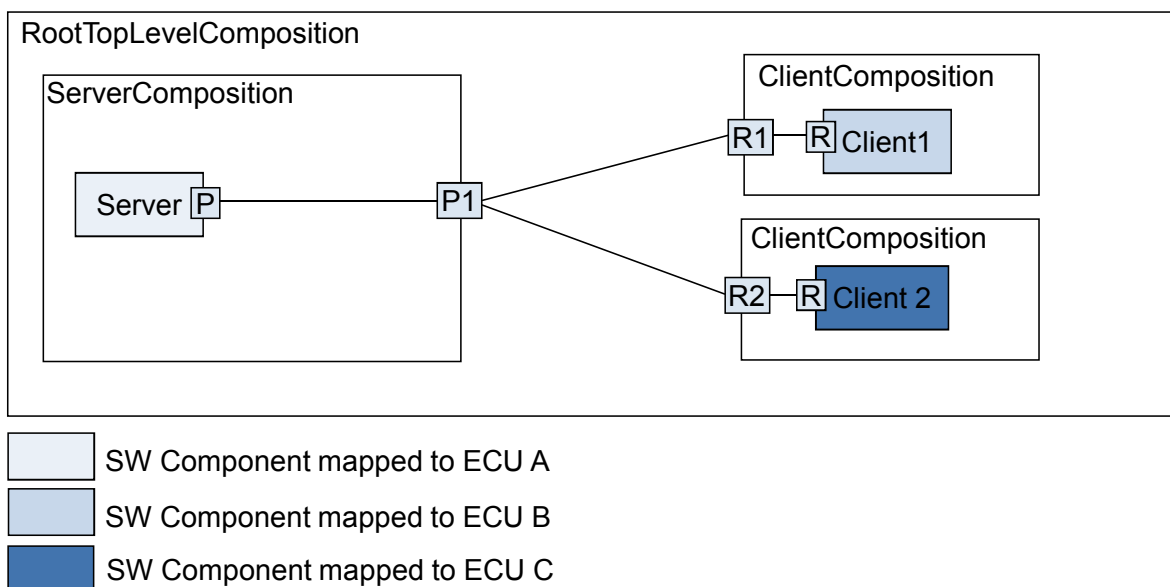


Figure 12.4: Client Server Scenario 3

For the scenario described in figure 12.4 the following statements apply:

- [ClientServerToSignalMappings](#) for the provide-ports Server.P and ServerComposition.P1 are ambiguous and thus `[constr_3243]` exists to forbid this situation.
- [ClientServerToSignalMappings](#) are permitted for ClientComposition.R1/ClientComposition.R2 and Client1.R/Client2.R (client-side) since there is no ambiguity.

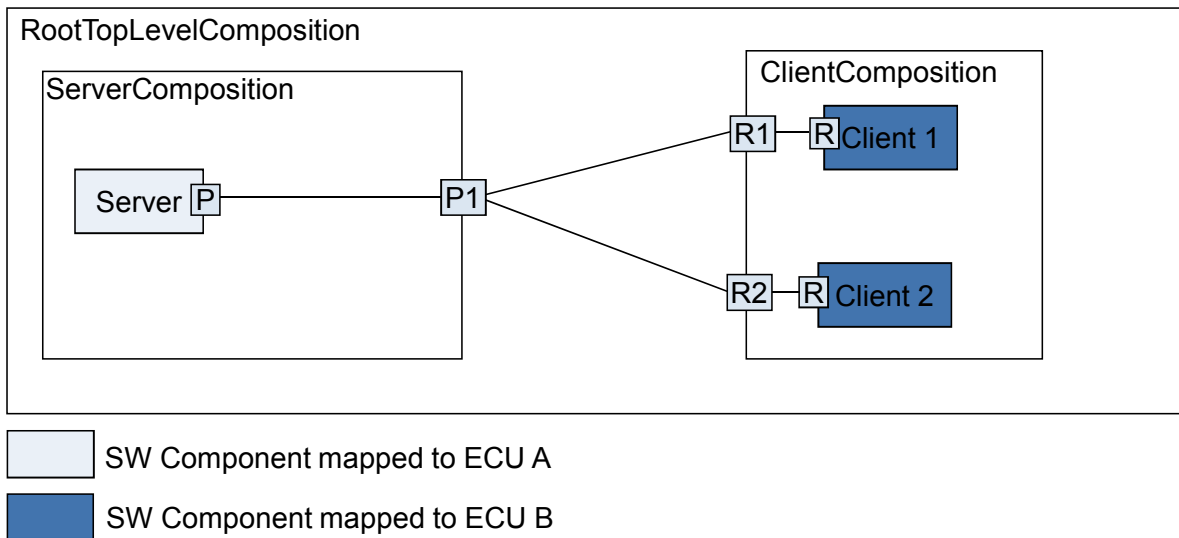


Figure 12.5: Client Server Scenario 4

For the scenario described in figure 12.5 the following statements apply:

- `ClientServerToSignalMappings` for the provide-ports `Server.P` and `ServerComposition.P1` are not ambiguous (since although there is fork in the communication path, both sub-paths end up at the same ECU).
- `ClientServerToSignalMappings` are permitted for `ClientComposition.R1/ClientComponsonion.R2` and `Client1.R/Client2.R` (client-side) or for `ServerComposition.P1/ServerComposition.P2` (provider-side) since there is no ambiguity.

[TPS_SYST_01145] PortInterfaceMappings in the ECU Extract [In the `System` with `category` `ECU_EXTRACT` the missing `PortInterfaceMappings` on the complementary side needs to be supplemented to `DelegationSwConnectors`.]()

Figure 12.6 shows how the `System` with `category` `ECU_EXTRACT` for ECU A of the SW composition that is defined in figure 11.3 would look like: Only those `SwComponentPrototypes` are included that are mapped to ECU A. The hierarchy present in the `System` with `category` `SYSTEM_DESCRIPTION` has been flattened into `CompositionSwComponentType` 'EcuAFlatView', including newly created `SwComponentPrototype` 'A1E', 'A2E', 'B1E' and 'C1E' for the component instances mapped to ECU A.

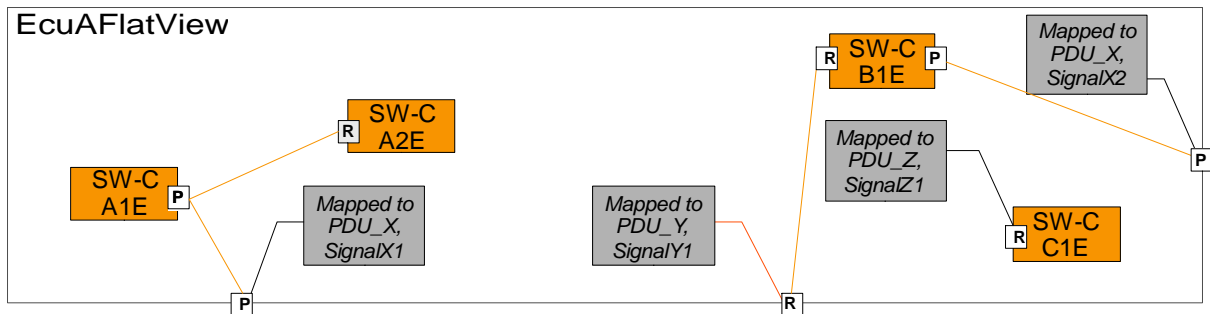


Figure 12.6: Example ECU extract for ECU A of above introduced composition

The `SwConnectors` to the outside ports (ECUFlatView composition ports) and `SwConnectors` that represent intra-ECU communication (in our example, only 'A1E' to 'A2E') are included. The `VariableDataPrototypes` and `ClientServerOperations` in the outside ports are mapped to `SystemSignals`. This `DataMapping` and the communication description is used to identify the source/destination of that data.

Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

The problem that new mapping rules have to be added arises with the mapping to 'PDU_Y', 'SignalY1': Since SW component 'A3', which was referenced in the original mapping, is no longer included, the `DataMapping` needs a new `VariableDataPrototype` in a `PortPrototype` to reference to. In the example, the data of the required port of 'B1E' is referenced, so that the ECU generator has the information that 'B1E' receives the data via 'PDU_Y'.

12.2.4 Port Groups

A `SwComponentType` can optionally define `PortGroups` which allow to group `PortPrototypes` according to logical criteria, e.g. according to shared communication resources (see [5]). A `PortGroup` of a `CompositionSwComponentType` can be linked to "inner" `PortGroups` of the aggregated `SwComponentPrototypes`. Since the main purpose of this grouping is to configure the behavior of mode managers on an `EcuInstance`, this information shall be preserved and broken down into the `System` with `category` `ECU_EXTRACT`.

The resulting `CompositionSwComponentType` in the ECU flat view will contain a set of `PortGroups` which refer to the linked inner port groups of the `SwComponentPrototypes` with `AtomicSwComponentType`. To get to this result, the following steps shall be applied in the extraction process:

- Recursively ignore all `PortGroups` in `CompositionSwComponentTypes` in the hierarchical structure, which are not linked to any inner groups to be mapped on this `EcuInstance`.

- In the remaining structure of linked `PortGroups` find out the top level `PortGroups` (i.e. which are not referred by any higher level `PortGroup` on this `EcuInstance`) and put an element representing each top level `PortGroup` into the `CompositionSwComponentType` of the ECU flat view. This can result in name conflicts, which should be resolved by a suitable algorithm.
- Link these top level `PortGroups` to the inner `PortGroups` of the atomic component instances of the flat view according to the links found in the hierarchical structure. Naturally, the top level `PortGroups` in the ECU flat view are not directly referring any `PortPrototypes` and due to the first step they should be linked to at least one inner `PortGroup`.
- The `PortGroups` in `SwComponentPrototypes` with an `AtomicSwComponentType` on the `EcuInstance` should be unchanged.

12.2.5 Service Needs

Each software component might need services which are provided by the ECU Basic Software through AUTOSAR Services. `ServiceNeeds` are used to provide detailed information what the software component expects from the AUTOSAR Services when integrated on an actual ECU (see `SWComponentTemplate` [5] for more details). If an ECU Extract is created the following rules apply to the existing `ServiceNeeds`:

[constr_3068] `DoIpPowerModeStatusNeeds` in the category `ECU_EXTRACT` [If and only if `DoIP` (i.e. any of the subclasses of `DoIpServiceNeeds` are present) is used on an Ecu then the `DoIpPowerModeStatusNeeds` shall exist exactly once in a `System` of category `ECU_EXTRACT`.]()

[constr_1265] `DoIpGidSynchronizationNeeds` can only exist once per `ECU_EXTRACT` [Within the context of one `System` of category `ECU_EXTRACT`, there can only be at most one `DoIpGidSynchronizationNeeds`.]()

[constr_1266] `DoIpGidNeeds` can only exist once per `ECU_EXTRACT` [Within the context of one `System` of category `ECU_EXTRACT`, there can only be at most one `DoIpGidNeeds`.]()

[constr_1267] `DoIpActivationLineNeeds` can only exist once per `ECU_EXTRACT` [Within the context of one `System` of category `ECU_EXTRACT`, there can only be at most one `DoIpActivationLineNeeds`.]()

[constr_3083] Exactly one `AtomicSwComponentType` on an `EcuInstance` may use `GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange` [The Dem only supports exactly one `AtomicSwComponentType` using `GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange` on one `EcuInstance`.]()

[constr_3084] Service port in the role `PowerTakeOff` [Within the context of one `EcuInstance`, there can only be one service port that uses the role `PowerTakeOff` in the `RoleBasedPortAssignment.role`.]()

[constr_3085] Service port in the role CallbackDCMRequestServices [Within the context of one `EcuInstance`, there can only be one service port that uses the role `CallbackDCMRequestServices` in the `RoleBasedPortAssignment.role`.]()

12.3 Communication

In explaining how `SystemSignals` are handled in the `System` with `category ECU_EXTRACT`, Section 12.2.3 touched on the topic of inter-ECU Communication. However, in order to enable the ECU Configuration of the COM-Stack, the relevant information of all layers of the AUTOSAR COM-Stack needs to be present in the `System` with `category ECU_EXTRACT`, including the central Communication classes `ISignal`, `Pdu` and `Frame`.

The above mentioned Communication elements have dependencies on each other, for ordinary COM-communication this means:

- `Frames` are assembled from one or more `Pdus`.
- `ISignalIPdus` carry their information in form of `ISignals`.
- `ISignals` as interaction points between RTE and COM refer to `SystemSignals`.

Note that the above list is not complete; TP and NM require additional elements. However, for the sake of clarity the following paragraphs describes the standard use case of a direct Signal-based communication between two `EcuInstances`. Once the handling of this case is understood, the additional model elements as `NPdu`, `NmPdu`, `SystemSignalGroup` etc. can be handled following the same basic principles.

For the `System` with `category ECU_EXTRACT` only the ECU-relevant subset of information present in the system-wide communication is to be considered. In order to establish this set of information, the dependencies in the list above are being followed.

12.3.1 Frame

In a complete `System` with `category SYSTEM_DESCRIPTION`, every outside communication of an `EcuInstance` will either be associated with an outgoing or and incoming `Frame`. The exact number and types of `Frames` to be received or sent by an `EcuInstance` is determined by the Communication Matrix (Chapter 6).

According to the selection rules for the Topology (12.1), the `System` with `category ECU_EXTRACT` contains all `FrameTriggerings` associated with `Frames` that are of any interest to the `EcuInstance`: If a particular `FrameTriggering` refers to a `FramePort` of type 'out' the associated `Frame` is to be sent by the `EcuInstance`, if it refers to an 'in' port the `Frame` is to be received. Therefore, the following selection rule applies:

- The `System` with `category` `ECU_EXTRACT` shall contain all `Frame` elements which are referenced by any included `FrameTriggering`.

12.3.2 PDU

`Frames` are assembled from one or more `Pdus`. In order to include all required `Pdu` elements, the following selection criteria apply:

- The `System` with `category` `ECU_EXTRACT` shall contain all `Pdu` elements which are referenced by any included `Frame`'s `PduToFrameMapping`.
- The `System` with `category` `ECU_EXTRACT` shall contain all `Pdu` elements which are referenced by any included `PduTriggering`.
- For multiplexed `Pdus`, additionally all `ISignalIPdus` referenced by the `MultiplexedIPdu`'s static and dynamic parts need to be included.

The second criterion is e.g. required in a pure post-build configuration scenario, where the frame-layout may not be completed at the time of `System` with `category` `ECU_EXTRACT` creation.

12.3.3 ISignals and ISignalGroups

`ISignalIPdus` carry their information in form of `ISignals` or `ISignalGroups`. In order to include all required `ISignal` and `ISignalGroup` elements, the following selection criteria apply:

- The `System` with `category` `ECU_EXTRACT` shall contain `ISignal` elements which are referenced by included `ISignalIPdu`'s `ISignalToIPduMapping`. One exception are Pdu Gateways. Signal definitions that are not directly relevant for `Gateways` in case that the `Pdu` is routed as a whole (Pdu Routing) shall be omitted. See Section 12.3.5 for more details.
- The `System` with `category` `ECU_EXTRACT` shall contain all `ISignal` elements which are referenced by any included `ISignalTriggering`.
- The `System` with `category` `ECU_EXTRACT` shall contain `ISignalGroup` elements which are referenced by included `ISignalIPdu`'s `ISignalToIPduMapping`. One exception are Pdu Gateways. Signal Group definitions that are not directly relevant for `Gateways` in case that the `Pdu` is routed as a whole (Pdu Routing) shall be omitted. See Section 12.3.5 for more details.
- The `System` with `category` `ECU_EXTRACT` shall contain all `ISignalGroup` elements which are referenced by any included `ISignalTriggering`.

Like in the case of the `Pdu` inclusion rules, the second and fourth criterion is required in scenarios with incomplete `Pdu` modeling due to post-build configurability of the communication matrix.

12.3.4 SystemSignal and SystemSignalGroup

Whereas the rules specified in Section 12.2.3 for the inclusion of `SystemSignal` comprise all `SystemSignals` that are being used by the Software Components in the ECU, the inclusion rules above stated for `ISignalIPdus` and `ISignals` may require the inclusion of additional `SystemSignals`. Also, strictly speaking both `SystemSignals` and `SystemSignalGroup` need to be considered. The complete inclusion rules for `SystemSignals` and `SystemSignalGroups` are:

- The `System` with `category` `ECU_EXTRACT` shall contain all `SystemSignals` and `SystemSignalGroup` elements which are referenced by any included `DataMapping`.
- The `System` with `category` `ECU_EXTRACT` shall contain all `SystemSignal` elements which are referenced by any included `ISignal`.
- The `System` with `category` `ECU_EXTRACT` shall contain all `SystemSignalGroup` elements which are referenced by any included `ISignalGroup`.

In addition on the receiving `EcuInstance` the following cases exist:

- only one `SystemSignal` out of the transmitted `SystemSignalGroup` is received: no `SystemSignalGroup` is required in the Ecu Extract of the receiving `EcuInstance`.
- more than one but not all `SystemSignals` out of the transmitted `SystemSignalGroup` are received: new `SystemSignalGroup` shall be created in the `System` with `category` `ECU_EXTRACT` of the receiving `EcuInstance` containing the received `SystemSignals`.
- all `SystemSignals` out of the transmitted `SystemSignalGroup` are received: the original `SystemSignalGroup` shall be taken over to the `System` with `category` `ECU_EXTRACT` of the receiving `EcuInstance`.

12.3.5 Gateways

Gateways that refer the `EcuInstance` shall be included in the `System` with `category` `ECU_EXTRACT`. The complete inclusion rules for `Gateways` are:

- The `System` with `category` `ECU_EXTRACT` shall contain all `FrameMapping` elements that are aggregated by the `Gateway` element.
- The `System` with `category` `ECU_EXTRACT` shall contain all `IPduMapping` elements that are aggregated by the `Gateway` element.
- The `System` with `category` `ECU_EXTRACT` shall contain all `ISignalMapping` elements that are aggregated by the `Gateway` element.

- `ISignal` definitions that are not directly relevant for the `Gateway` in case that the `Pdu` containing these `ISignals` is routed as a whole (Pdu Routing) shall be omitted .
- `ISignalGroup` definitions that are not directly relevant for the `Gateways` in case that the `Pdu` containing these `ISignalGroups` is routed as a whole (Pdu Routing) shall be omitted .

12.3.6 TP configuration

The TP-configuration element `TpConfig` and all its associated elements shall be included into the `System` with `category` `ECU_EXTRACT` if the `EcuInstance` has an `TpAddress` configured in this `TpConfig`.

12.3.7 NM configuration

The Nm configuration part of the `System` with `category` `ECU_EXTRACT` shall include the `NmEcu` that references the included `EcuInstance`. In addition a `NmCoordinator` composed by this `NmEcu` shall be included. Furthermore any `NmNode` referenced by the `NmCoordinator` shall be included. For each included `NmNode` the composing `NmCluster` shall be included. For each included `NmCluster` the composing `NmConfig` shall be included.

12.4 Naming Issues

[TPS_SYST_05015] Naming conventions [The definition of naming conventions may facilitate the avoidance of name clashes to the further degree. However, these naming conventions can only be defined on the model level and the System Template does not define any specific naming conventions.] ([RS_SYST_00053](#))

Please note that a detailed information about mechanisms to resolve naming conflicts is given in [4]: [TR_METH_03005], [TR_METH_03006], [TR_METH_03007], [TR_METH_03008], [TR_METH_03009], [TR_METH_03010].

12.4.1 Package Structure

As detailed in the sections above, extracting information from the `System` with `category` `SYSTEM_DESCRIPTION` into an `System` with `category` `ECU_EXTRACT` is a non-trivial transformation: While some of the model elements are simply copied verbatim into the `System` with `category` `ECU_EXTRACT`, it is additionally necessary to create new elements reducing parts of system-wide structures, most noticeably in flattening of the hierarchical VFB view to the ECU Flat View.

All such elements being created or modified in the process of generating the `System` with `category` `ECU_EXTRACT` shall reside in the same `ARPackage`. In order to avoid namespace conflicts with existing elements, the package shall exclusively be used for this purpose.

By creating derivation elements from elements originally contained in the `System` with `category` `SYSTEM_DESCRIPTION` package structure, duplications of names may occur. This kind of name clashes shall be resolved by a suitable naming algorithm (see section 12.4.3).

All Elements that are taken over from the `System` with `category` `SYSTEM_DESCRIPTION` unchanged (e.g. `AtomicSwComponentType`, `PortInterface`, `ApplicationDataType`, `EcuInstance`, `CommunicationCluster`) shall remain in their original packages.

`ARElements` not used in the `System` with `category` `ECU_EXTRACT` shall not be copied to the ECU Extract XML file.

In more detail, `ARPackages` taken over from `System` with `category` `SYSTEM_DESCRIPTION` will not be altered by the ECU extraction process, except that some `ARElements` will not be included in the actual XML file of the extract: `ARElements` which exist in the `System` with `category` `SYSTEM_DESCRIPTION` but have been stripped for the `System` with `category` `ECU_EXTRACT` are not actually deleted from their `ARPackage`, but merely are skipped in the XML file forming the extract. Note that having such a partial view on an `ARPackage` doesn't break the original `ARPackage` definition because the composition of `PackageableElement`, responsible for adding `ARElements` to `ARPackage`, is stereotyped `<<splitable>>`; this means several XML files can contribute to an `ARPackage`, or in case of the ECU Extract an AUTOSAR description file may contain only a subset of the complete `ARPackage`.

12.4.2 Naming of Measurement and Calibration Data

The software component descriptions provide several means to declare data prototypes which have to be available for measurement and calibration (MCD) tools on the `EcuInstance`. Together with the `System` with `category` `ECU_EXTRACT` it is required to provide a list of references to the description of these data for further processing in the scope of the `EcuInstance`. In addition, the MCD tools need a unique name for each instance of such a data prototype. Since the data descriptions are part of the nested composition structure and are contained in reusable types (components or port interfaces), the system description itself does in general not provide unique names for those.

This means, providing such a list with references and unique names for MCD data is also a task of the ECU extractor tool. This list is part of the artifact ECU Flat Map, which is further explained below.

12.4.3 Naming of Derived Elements

When performing the extract process, name clashes may occur, necessitating a naming scheme for elements derived in ECU generation: By flattening the Software Composition hierarchy all component instances present on the considered `EcuInstance` are put in one ECU-wide software composition. Name clashes may occur for the following reasons:

1. `SwComponentPrototypes` taken from different Software Compositions are allowed to have identical short names in the hierarchical structure. As all `SwComponentPrototypes` will be located in the same ECU Flat View, the original name spaces separation no longer exists.
2. Multiple instances of the same `CompositionSwComponentType` are mapped to an `EcuInstance`: In this case, duplicates of all contained `SwComponentPrototypes` will be placed next to each other in the ECU flat composition.
3. The two mechanisms just mentioned may also lead to name clashes in `AutosarDataPrototypes` if their names shall be used as MCD data names. In addition, reuse of a `PortInterface` can also lead to name clashes if it provides data elements to be used by MCD.
4. The setup of `PortGroups` in the ECU flat view can result in name clashes, because two port groups originating from different component types (i.e. different name spaces) may be aggregated within the flat view.

Therefore the `System` with `category` `ECU_EXTRACT` generator shall take care that all elements derived or created during the extraction process have unique short names. These unique names shall be created in an initial step of the extraction process which leads to the creation of an initial ECU Flat Map. Some ways to satisfy this requirement may be:

- Use globally unique identifiers (GUID) for generating short names.
- Add a number to the original name; if done consistently the flat map approach makes this reproducible.
- Expand the name recursively by the names of the containing elements (e.g. compositions) until it is unique.
- Allow human interaction (this may be combined with an initially proposed name expansion).

The creation of a new short name is compulsory only if otherwise a clash would occur.

[constr_2025] Uniqueness of `symbol` attributes [With the exception of `RunnableEntities` that are subject to [constr_1234] (`RunnableEntities` owned by `NvBlockSwComponentTypes`), in the context of a single `EcuInstance` the values of the `RunnableEntity.symbol` in combination with the attribute `symbol` of the meta-class `SymbolProps` owned by `AtomicSwComponentType` of all deployed `RunnableEntities` shall be unique such that no two (or more) combinations of `RunnableEntity.symbol` and

the `symbol` of the meta-class `SymbolProps` owned by `AtomicSwComponentType` in the role `symbolProps` share the same value.>()

12.4.4 Re-use of short names assigned in previous iterations

As described in the previous section, potential name clashes during ECU extraction shall be avoided by assigning unique names to the elements specifically created for the `System` with `category` `ECU_EXTRACT` and for the list of MCD data per `EcuInstance`. Considering the use case of iterative development (also see Section 12.5), the same names shall be assigned to existing elements in consecutive iterations. Elements which have been modified or newly introduced between two ECU extract iterations shall not use an existing short name. Additionally, the ECU extractor tool shall not re-use any short name used in any iteration from previous development phases if the meaning of the element is not exactly the same (i.e. the element's back reference into the System Configuration Description is not the same.)

12.5 ECU Extract in subsequent Cycles of Iterative Development

12.5.1 Traceability of model elements created in ECU Extract

For development scenarios in real life projects iterative development shall be supported.

The following use case shall be considered:

Changes in the `System` with `category` `SYSTEM_DESCRIPTION` require the recreation of an `System` with `category` `ECU_EXTRACT`. In the successive re-run of ECU configuration, ECU configuration parameters which were configured based on the previous `System` with `category` `ECU_EXTRACT` need to be maintained for those parts in the `System` with `category` `ECU_EXTRACT` that didn't change between iterations.

Consequently, there are two requirements on the extraction process:

- Elements that are present in both versions of the `System` with `category` `SYSTEM_DESCRIPTION` shall not change their short names between the two ECU Extracts either.
- If changes between the two versions of the `System` with `category` `SYSTEM_DESCRIPTION` lead to the creation of new model elements in the `System` with `category` `ECU_EXTRACT`, then these newly created elements shall have new names that have not been used in previous iterations of the `System` with `category` `ECU_EXTRACT`. (See also Section 12.4.4).

In order to fulfill these requirements, a back-tracing of the relevant model elements in the `System` with `category` `ECU_EXTRACT` to their counterparts in the `System` with `category` `SYSTEM_DESCRIPTION` shall be established. Based on these back

references, short names shall consistently be re-used in iterations. Relevant elements are all those which potentially have been modified in the extraction process.

All back-tracing references are collected in one central table per *System* with *category* ECU_EXTRACT based on the meta-class *FlatMap*. This table collects “instance” entries for each Ecu Extract element that is being created in the *System* with *category* ECU_EXTRACT transformation and for each MCD data object that has to be available in the *EcuInstance*. These entries are called *FlatInstanceDescriptor*.

Each mapping entry owns two references per mapped element, one reference pointing to the target element in the *System* with *category* ECU_EXTRACT, the other one pointing to the origin in the *System* with *category* SYSTEM_DESCRIPTION. Both of these references are deep “instance” references, requiring a tuple of context/target description.

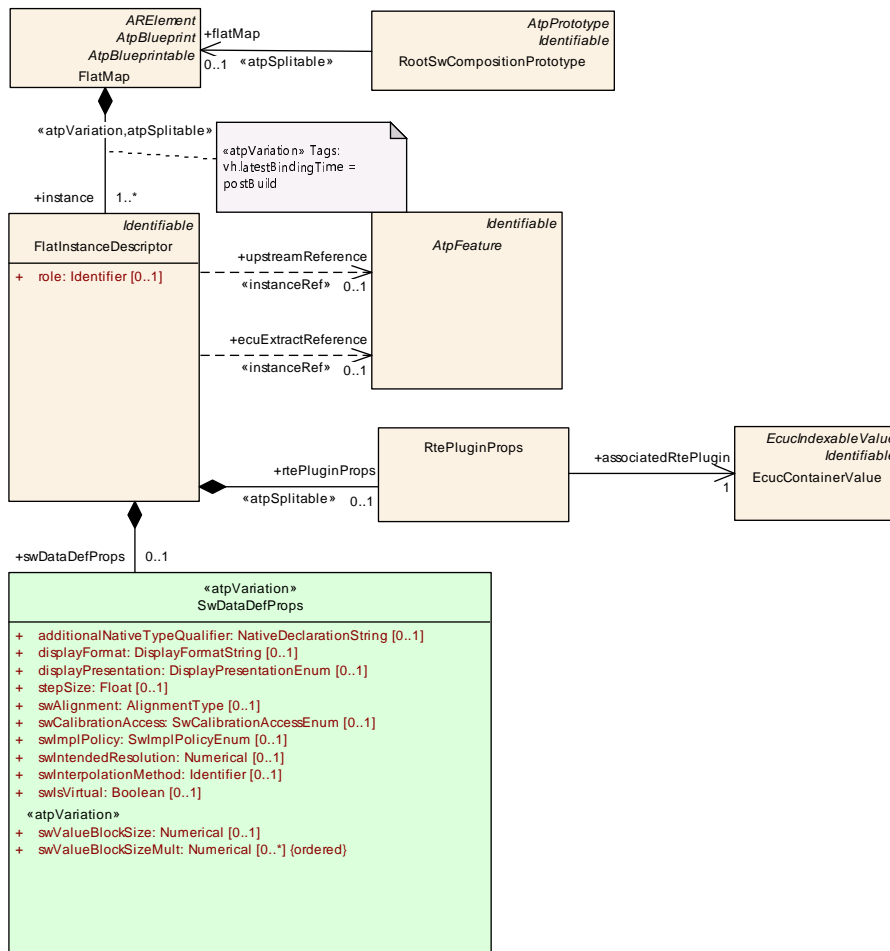


Figure 12.7: Flat Map (CommonStructure: FlatMap)

Class	FlatMap			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	<p>Contains a flat list of references to software objects. This list is used to identify instances and to resolve name conflicts. The scope is given by the RootSwCompositionPrototype for which it is used, i.e. it can be applied to a system, system extract or ECU-extract.</p> <p>An instance of FlatMap may also be used in a preliminary context, e.g. in the scope of a software component before integration into a system. In this case it is not referred by a RootSwComposition Prototype.</p> <p>Tags:atp.recommendedPackage=FlatMaps</p>			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
instance	FlatInstanceDescriptor	1..*	aggr	<p>A descriptor instance aggregated in the flat map.</p> <p>The variation point accounts for the fact, that the system in scope can be subject to variability, and thus the existence of some instances is variable.</p> <p>The aggregation has been made splittable because the content might be contributed by different stakeholders at different times in the workflow. Plus, the overall size might be so big that eventually it becomes more manageable if it is distributed over several files.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>

Table 12.1: FlatMap

Class	FlatInstanceDescriptor			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	<p>Represents exactly one node (e.g. a component instance or data element) of the instance tree of a software system. The purpose of this element is to map the various nested representations of this instance to a flat representation and assign a unique name (shortName) to it.</p> <p>Use cases:</p> <ul style="list-style-type: none"> Specify unique names of measurable data to be used by MCD tools Specify unique names of calibration data to be used by MCD tool Specify a unique name for an instance of a component prototype in the ECU extract of the system description <p>Note that in addition it is possible to assign alias names via AliasNameAssignment.</p>			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
ecuExtract Reference	AtpFeature	0..1	iref	<p>Refers to the instance in the ECU extract. This is valid only, if the FlatMap is used in the context of an ECU extract.</p> <p>The reference shall be such that it uniquely defines the object instance. For example, if a data prototype is declared as a role within an SwcInternalBehavior, it is not enough to state the SwcInternalBehavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying</p>





Class	FlatInstanceDescriptor			
				<p style="text-align: center;">△</p> <p>instance of the component prototype and the Atomic SoftwareComponentType, which is referred by the particular SwcInternalBehavior.</p> <p>Tags:xml.sequenceOffset=40</p>
role	Identifier	0..1	attr	<p>The role denotes the particular role of the downstream memory location described by this FlatInstanceDescriptor.</p> <p>It applies to use case where one upstream object results in multiple downstream objects, e.g. ModeDeclaration GroupPrototypes which are measurable. In this case the RTE will provide locations for current mode, previous mode and next mode.</p>
rtePluginProps	RtePluginProps	0..1	aggr	<p>The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in.</p> <p>Stereotypes: atpSplittable Tags:atp.Splitkey=rtePluginProps</p>
swDataDef Props	SwDataDefProps	0..1	aggr	<p>The properties of this FlatInstanceDescriptor.</p>
upstream Reference	AtpFeature	0..1	iref	<p>Refers to the instance in the context of an "upstream" descriptions, which could be the system or system extract description, the basic software module description or (if a flat map is used in preliminary context) a description of an atomic component or composition. This reference is optional in case the flat map is used in ECU context.</p> <p>The reference shall be such that it uniquely defines the object instance in the given context. For example, if a data prototype is declared as a role within an SwcInternal Behavior, it is not enough to state the SwcInternal Behavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying the instance of the component prototype that contains the particular instance of Swc InternalBehavior.</p> <p>Tags:xml.sequenceOffset=20</p>

Table 12.2: FlatInstanceDescriptor

[TPS_SYST_01000] [FlatInstanceDescriptor](#) roles [If a ModeDeclarationGroupPrototype is measurable the FlatMap shall contain three entries where the particular roles are set to

- CURRENT_MODE specifies the [FlatInstanceDescriptor](#) applicable for current mode value of the [ModeDeclarationGroupPrototype](#)
- PREVIOUS_MODE specifies the [FlatInstanceDescriptor](#) applicable for previous mode value of the [ModeDeclarationGroupPrototype](#)
- NEXT_MODE specifies the [FlatInstanceDescriptor](#) applicable for next mode value of the [ModeDeclarationGroupPrototype](#)

Please note that these entries may exist in a [FlatMap](#) even if the [ModeDeclarationGroupPrototype](#) is not measurable.]([RS_SYST_00003](#), [RS_SYST_00027](#))

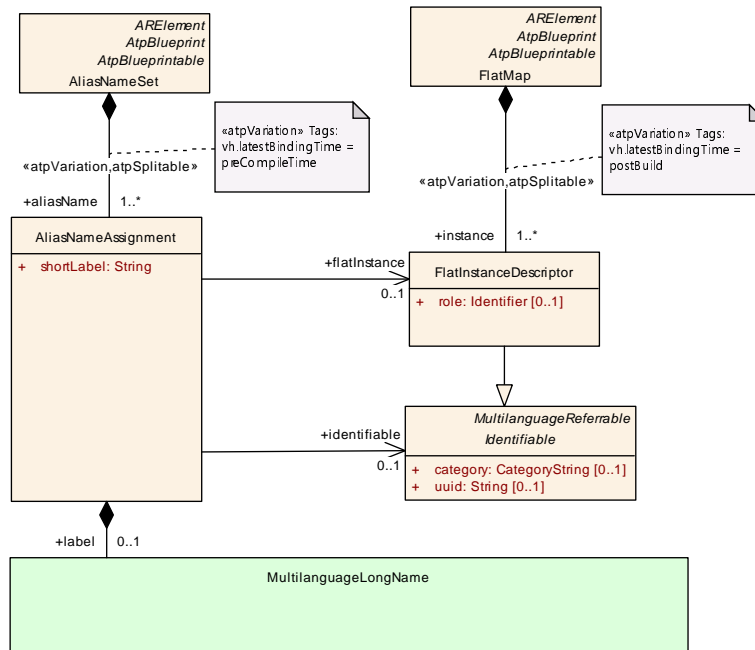


Figure 12.8: Alias Name Assignment (CommonStructure: AliasNameAssignment)

Class	AliasNameSet			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	This meta-class represents a set of AliasNames. The AliasNameSet can for example be an input to the A2L-Generator. Tags: atp.recommendedPackage=AliasNameSets			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
aliasName	AliasNameAssignment	1..*	aggr	AliasNames contained in the AliasNameSet. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortLabel vh.latestBindingTime=preCompileTime

Table 12.3: AliasNameSet

Class	AliasNameAssignment			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	This meta-class represents the ability to associate an alternative name to a flat representations or an Identifiable. The usage of this name is defined outside of AUTOSAR. For example this name can be used by MCD tools or as a name for component instances in the ECU extract. Note that flatInstance and identifiable are mutually exclusive.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note





Class	AliasNameAssignment			
flatInstance	FlatInstanceDescriptor	0..1	ref	Assignment of a unique name to a flat representation. Tags: xml.sequenceOffset=60
identifiable	Identifiable	0..1	ref	Assignment of a unique name to an Identifiable. Tags: xml.sequenceOffset=50
label	MultilanguageLong Name	0..1	aggr	This represents an "Alias LongName". Tags: xml.sequenceOffset=20
shortLabel	String	1	attr	This attribute represents the alias name. It is modeled as string because the alias name is used outside of AUTOSAR and therefore no naming conventions can be applied within AUTOSAR. Tags: xml.sequenceOffset=10

Table 12.4: AliasNameAssignment

During the ECU extraction process, the ECU [FlatMap](#) will be processed in the following steps:

1. Create the entries [shortName](#) and [upstreamReference](#) of the [FlatMap](#) or, if a previous version exists, try to reuse them. Resolve name conflicts.
2. Generate the ECU Software Composition.
3. Create the entries [ecuExtractReference](#) of the ECU [FlatMap](#).

More details are define be the AUTOSAR methodology, see [4]. The methodology also allows to have a [FlatMap](#) for the whole system. This System [FlatMap](#) can be created and maintained independently from the ECU extraction process, but can be used as an input for the creation of the ECU [FlatMap](#).

[constr_3378] Maximal one [AliasNameAssignment](#) allowed per [FlatInstanceDescriptor](#) [In a given instance of [AliasNameSet](#) in the bound system there shall be at most one [aliasName](#) per [FlatInstanceDescriptor](#).]()

12.5.2 Mapping of AUTOSAR attributes to ASAM ASAP2

With the MC Support information AUTOSAR builds a bridge to tools processing ASAM ASAP2 files. In order to support the interoperability of converter tools the following mapping of AUTOSAR attributes to ASAM ASAP2 [19] (also known as "A2I" respectively "ASAM MCD 2MC") is recommended:

- If the [FlatInstanceDescriptor](#) references DataPrototypes:

```
FlatInstanceDescriptor.shortName ->
MEASUREMENT Name
CHARACTERISTIC Name
```

`FlatInstanceDescriptor.(longName + desc |upstreamReference.desc) ->`
 MEASUREMENT LongIdentifier
 CHARACTERISTIC LongIdentifier

`AliasNameAssignment.shortLabel ->`
 MEASUREMENT [-> DISPLAY_IDENTIFIER]
 CHARACTERISTIC [-> DISPLAY_IDENTIFIER]

`AliasNameAssignment.label(if provided) +`
`FlatInstanceDescriptor.(desc |upstreamReference.desc) ->`
 MEASUREMENT LongIdentifier
 CHARACTERISTIC LongIdentifier

- If `AliasNameAssignment` references a `SwSystemconstant`:

`AliasNameAssignment.shortLabel ->`
 SYSTEM_CONSTANT -> Name for `SwSystemconstants`

- If `AliasNameAssignment` references a `Unit`:

`AliasNameAssignment.shortLabel ->`
 UNIT -> Name for Units

12.5.3 Assigning communication graphs to RTE Implementation Plug-Ins

When `RTE Implementation Plug-Ins` are used to modularize the RTE implementation, it's required to decide which communication graphs are implemented by the RTE or by a specific `RTE Implementation Plug-In`. Thereby an `RTE Implementation Plug-In` is a part of the overall RTE implementation which is not provided by the RTE Generator but from an additional source (e.g. a Plug-In Generator or a manually implemented source code).

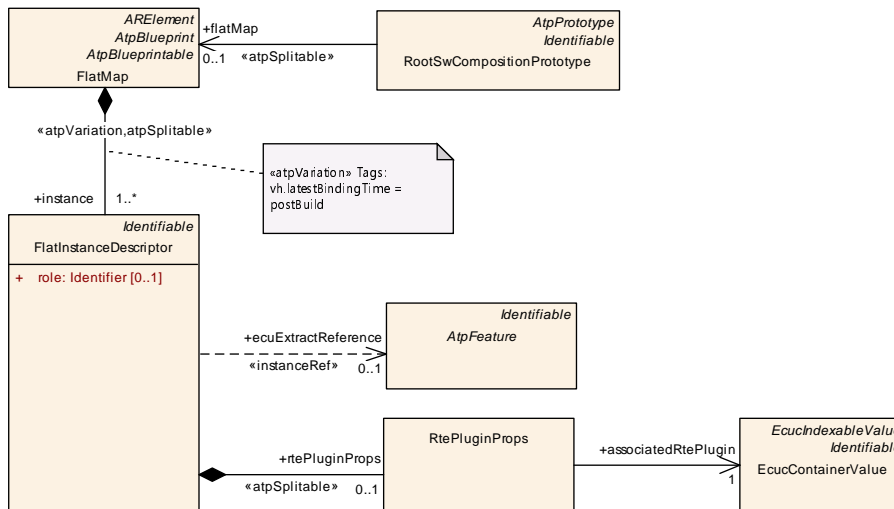


Figure 12.9: ECU Flat Map and `rtePluginProps`

Class	RtePluginProps			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
associatedRtePlugin	EcucContainerValue	1	ref	This associates a communication graph to a specific RTE Implementation Plug-in.

Table 12.5: RtePluginProps

This assignment is described with the [FlatInstanceDescriptor.rtePluginProps](#), where the [RtePluginProps.associatedRtePlugin](#) references the [EcucContainerValue](#) representing the identity of an RTE Implementation Plug-In.

Assigning an communication graphs has following underlying semantic:

[TPS_SYST_02197] Assigning communication graphs to RTE Implementation Plug-Ins [The [FlatInstanceDescriptor.ecuExtractReference](#) points to an instance of a [VariableDataPrototype](#) and the [FlatInstanceDescriptor.rtePluginProps.associatedRtePlugin](#) references the [EcucContainerValue](#) which defines the identity of the RTE Implementation Plug-In. This assigns the full communication graph to the specific RTE Implementation Plug-Ins represented by according [EcucContainerValue](#).] ([RS_SYST_00057](#))

For instance the [FlatInstanceDescriptor.ecuExtractReference](#) points to instance of a [VariableDataPrototype](#) defined by the [AnyInstanceRef](#) using

- [contextElement: RootSwCompositionPrototype](#)
- [contextElement: SwComponentPrototype](#)
- [contextElement: PPortPrototype](#)
- [target: VariableDataPrototype](#)

According the AUTOSAR Meta-Model various further model elements are exist to describe the complete communication graph, for instance with the means of [AssemblySwConnectors](#), [SwComponentType.ports](#), [RunnableEntities](#) and [VariableAccesses](#). Nevertheless all such related elements of this communication graph are addressed by this single [FlatInstanceDescriptor](#) and all access of [ExecutableEntities](#) to this communication graph are handled by the associated RTE Implementation Plug-In.

[constr_3458] FlatInstanceDescriptor.rtePluginProps shall only reference a EcucContainerValue representing a RteRipsPlugin [[FlatInstanceDescriptor.rtePluginProps](#) shall only reference an [EcucContainerValue](#) which defines the identity of the RTE Implementation Plug-In. This requires that the according [EcucContainerValue](#)'s definition references a [EcucContainerDef](#) having a [destinationUri](#) set to `/AUTOSAR/EcucDestinationUriDefSets/RteRipsUriDefSet/RteRipsPlugin`]()

To support different work-flows the `FlatInstanceDescriptor.rtePluginProps` is defined as `<<atpSplittable>>`. Therefore it's possible to do the assignment of communication graphs immediately during the creation of the ECU Flat Map or in a second processing step after the ECU Flat Map is already created.

Further information is provided in the document SWS RTE [32] at which specific anchor points of an communication graphs the assignment shall be described.

Some further notes about the chosen modeling pattern

In general it is an unusual pattern, that a meta class not being related to ECU configuration references an `EcucContainerValue`. But the `FlatInstanceDescriptor` of the ECU Extract is in any case closely related to the configuration of the ECU. Furthermore in case of data conversion it's mandatory to provide for each different representation a `FlatInstanceDescriptor` for the communication graph. Further information about such configurations is provided in the document SWS RTE [32].

The alternative approach to describe the `RTE Implementation Plug-In` as meta class is not the right approach since only very few properties of `RTE Implementation Plug-Ins` are standardized. There is also no need to exchange information between different development parties about those properties. Due to this reason `RTE Implementation Plug-Ins` are described by the means of ECU Configuration elements.

The alternative approach to model the relationship between `FlatInstanceDescriptors` and the container which represents the `RTE Implementation Plug-In` with a mapping pattern was rejected due to the very high number of expected configuration elements.

12.6 Variant Handling in ECU Extract

The System Template supports the creation of variants in many of its model elements. Depending on the binding time, some of this variability may have been already resolved within the `System` with `category` `SYSTEM_DESCRIPTION` at the time of creating the `System` with `category` `ECU_EXTRACT`, and a cleanup step may have removed some of the complexity by removing the out-configured variability.

If however binding of a concrete variation condition happens in a later stage of the AUTOSAR methodology (e.g. during ECU Configuration or even post build), or if for other process reasons such a cleanup step is not applicable, the variability needs to be carried over to the `System` with `category` `ECU_EXTRACT`.

12.6.1 System Constants

In the AUTOSAR variant handling concept, `SwSystemconst` represents a variant selector which needs to have its value assigned latest at binding time of any expression

which refers to it. Such a value assignment may be done literally using a fixed value, or by specifying a formula, depending on the values of other variant selectors. The elements to do this are collected in a `SwSystemconstantValueSet`, aggregating individual value assignment expressions in the form of `SwSystemconstValue`.

In the `System` with `category` `ECU_EXTRACT`, all `SwSystemconst` elements are included that influence its variable content. In detail the following rules for the inclusion of `SwSystemconst` apply:

- `System` with `category` `ECU_EXTRACT` shall contain all `SwSystemconst` elements that are being referenced directly by variable elements contained in the `System` with `category` `ECU_EXTRACT`.
- Additionally, whenever a `SwSystemconst`'s value is assigned indirectly using an `SwSystemconstValue`'s `ConditionByFormula` expression, each `SwSystemconstValue` referred to in the assignment formula needs to be included, too. As such assignments may be nested in multiple levels, the whole directed acyclic graph of `SwSystemconst` elements influencing the `System` with `category` `ECU_EXTRACT` variability need to be included.

Additionally to the `SwSystemconst` elements also all relevant `SwSystemconstValue` assignments need to be included. As they are aggregated by `SwSystemconstantValueSet`, the whole Value Set is included whenever one of its `SwSystemconstValue` assignments is relevant for the `System` with `category` `ECU_EXTRACT`.

Note: Typically, the assignment of Variants ("Binding") will be done in a Variant Configuration work product, separate from the actual `System` with `category` `ECU_EXTRACT`. In this case, the relevant information from the Variant Configuration also needs to be extracted and delivered in combination with the `System` with `category` `ECU_EXTRACT`. From the model point of view it doesn't matter whether `System` with `category` `ECU_EXTRACT` and Variant Configuration are contained in the same file or in separate files.

12.6.2 Nested Whole/Part class variants

In case of flattening the hierarchical VFB view to the ECU flat view representation, the case may appear that one conditional `SwComponentPrototype` is nested within another `SwComponentPrototype` depending on another variance condition. As the resulting ECU flat view only has a flat representation of `SwComponentPrototypes`, such a double condition needs to be resolved to a single condition in the resulting `SwComponentPrototypes`.

In this case, the variation condition formula needs to be altered such that the two (or more) individual conditions are combined in a boolean AND function.

12.6.3 Multiple instances of calibration parameters in system scope

Use case: In complex systems the problem occurs that parameter values may depend on the configuration of the vehicle due to functional side effects. E.g. the calibration of a lambda sensor depends from the kind of transmission due to mechanical impacts (e.g due to additional / different curvatures in the exhaust pipe)

The difficulty is that those dependencies are typically detected after design of the software components and shall not change the software component design. Furthermore this is typical use case for post build variability since the ECU SW should not change due to environmental variability.

[TPS_SYST_02029] Multiple `ParameterDataPrototype` instances in an `EcuExtract` [It shall be possible to instruct the RTE Generator to provide various instances for a `ParameterDataPrototype` in the `System` with `category` `ECU_EXTRACT`. Therefore one `FlatInstanceDescriptor` per expected data instance has to point to the `ParameterDataPrototype` as an `atpTarget`.]()

[constr_3114] `FlatInstanceDescriptors` pointing to the same `ParameterDataPrototype` shall have different `postBuildVariantConditions` [`FlatInstanceDescriptors` that are pointing as an `atpTarget` to the same `ParameterDataPrototype` instance shall have different `postBuildVariantConditions`.]()

Note: When several instances of a `ParameterDataPrototype` are created it shall be ensured that at most one parameter instance is active in a post build variant.

[constr_3115] `FlatInstanceDescriptors` pointing to the same `ParameterDataPrototype` instance [When several `FlatInstanceDescriptors` point to the same `ParameterDataPrototype` instance as an `atpTarget` in the context of a `ParameterInterface` the different `FlatInstanceDescriptors` shall point to the `PPortPrototype` of the owning `ParameterSwComponentType`. In this case the `PPortPrototype` typed by the `ParameterInterface` is part of the context of the according `AnyInstanceRef`.]()

Please note that the individual `FlatInstanceDescriptors` are utilized to provide unique names for the MCD tool as well as individual `CalibrationParameterValues` typically refer to the `FlatInstanceDescriptors` to provide instance specific initialization values.

13 Supported special use-cases

The description means of the communication matrix in the System Template potentially support a variety of use-cases. Some combinations of description means are explicitly ruled-out by semantical constraints. But the remaining space for the possible descriptions is so huge, that certain use-cases are actually not supported by tool-vendors because they did not consider them. This chapter describes special use-cases that can be specified in the System Template in order to get a harmonized support by tools.

13.1 Support of sending / receiving same Can/Flexray Frame on same channel (Pdu Gateway Use-Case)

Description: The System Template supports the definition of a communication where the same Can/Flexray `FrameTriggering` is sent and received on the same `PhysicalChannel` of one Pdu Gateway `EcuInstance`.

Rationale: This use-case occurs in gateway `EcuInstances` which are used in several vehicle platforms.

Implementation: This usage shall be supported by defining one `Frame` and one `FrameTriggering` with different directions on the referenced `FramePorts` for the same `PhysicalChannel`. Also one `Pdu` and one `PduTriggering` with different directions on the referenced `IPduPorts` for the same `PhysicalChannel` shall be used.

Example: In figure 13.1 a sample network setup is shown. The ECU1 is designed to send the `Frame_X` on the `PhysicalChannel`. The ECU2, ECU3 and ECU4 do receive the information. But since ECU1 is optional, ECU4 is also designed to send the `Frame_X` on the network (in case ECU1 is not present). Please note that in in this example ECU4 is a gateway `EcuInstance` that is connected to an additional channel.

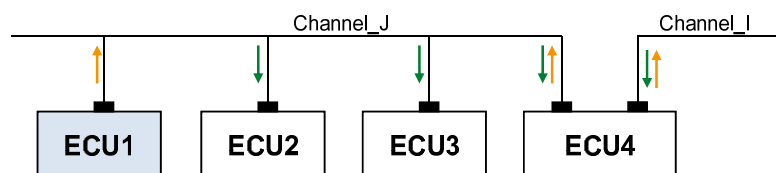


Figure 13.1: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the `Frame_X` and one `FrameTriggering` for the `PhysicalChannel` (figure 13.2). Each `EcuInstance` sending or receiving the `FrameTriggering` does define one `FramePort` per direction, thus for ECU4 there are two `FramePorts` defined.

For each `Pdu` mapped to the `Frame` there exists one definition for the `Pdu_X` and one `PduTriggering` for the `PhysicalChannel`. Each `EcuInstance` sending or receiving the `Pdu` does define one `IPduPort` per direction, thus for ECU4 there are two `IPduPorts` defined.

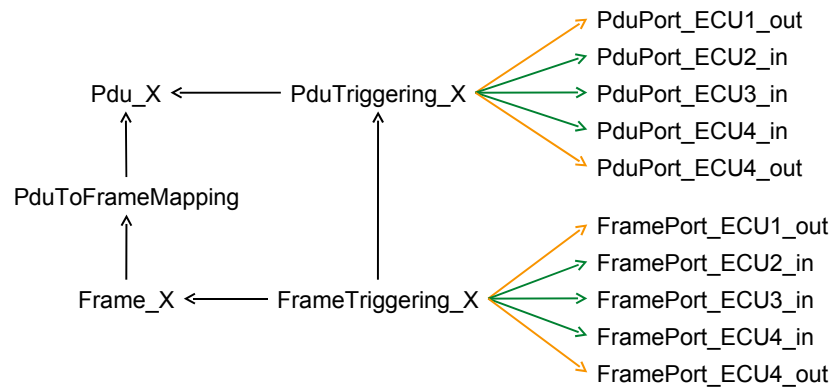


Figure 13.2: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent by the same Gateway ECU

In case a System Extract / ECU Extract is build, only the relevant `FramePorts` and `IPduPorts` for the corresponding `EcuInstance` are extracted. Especially in case an additional `EcuInstance` is designed to send and receive the same `Frame` all the other ECU extracts will not be affected by this change.

13.2 Support of sending / receiving same Can/Flexray Frame on same channel (bidirectional routing in COM)

Description: The System Template supports the definition of a communication where the same Can/Flexray `FrameTriggering` is sent and received on the same `PhysicalChannel` of one `EcuInstance` and the content of this `Frame` is processed by an Application.

Rationale: This use-case occurs in case of runtime variation where the same data is transmitted or received by the same ECU.

Implementation in a System Description: This use-case is supported with the following modelling:

- One `Frame` and one `FrameTriggering` with different directions on the referenced `FramePorts` for the same `PhysicalChannel` shall be defined.
- One `Pdu` and one `PduTriggering` with different directions on the referenced `IPduPorts` for the same `PhysicalChannel` shall be defined.
- One `ISignal` and one `ISignalTriggering` with different directions on the referenced `ISignalPorts` for the same `PhysicalChannel` shall be defined.

Please note that in case of a bidirectional routing on the `ISignal` level the COM Configuration (`ComIPdu`s) needs to be derived from the `PduTriggering` and from `IPduPorts`.

Example: In figure 13.3 a sample network setup is shown. The same data (`Frame_X`) is transmitted by Ecu4 and by Ecu1 (runtime variation). Ecu4 is designed to send and to receive the `Frame_X` on the network. For Ecu2 and Ecu3 it is transparent from which sender (Ecu1 or Ecu4) the data is transmitted.

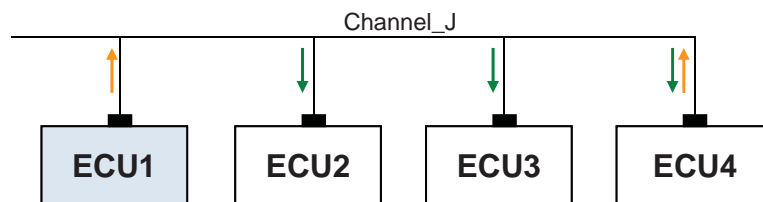


Figure 13.3: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the `Frame_X` and one `FrameTriggering` for the `PhysicalChannel` (figure 13.4). Each `EcuInstance` sending or receiving the `FrameTriggering` does define one `FramePort` per direction, thus for ECU4 there are two `FramePorts` defined.

For each `Pdu` mapped to the `Frame` there exists one definition for the `Pdu_X` and one `PduTriggering` for the `PhysicalChannel`. Each `EcuInstance` sending or receiving the `Pdu` does define one `IPduPort` per direction, thus for ECU4 there are two `IPduPorts` defined.

For each `ISignal` mapped to the `Pdu` there exists one definition for the `Signal_X` and one `ISignalTriggering` for the `PhysicalChannel`. Each `EcuInstance` sending or receiving the `ISignal` does define one `ISignalPort` per direction, thus for ECU4 there are two `ISignalPorts` defined.

Example 13.4 shows a System Description where only the `DataMapping` for the PPorts is defined. Please note that in the COM configuration a `ComIPdu` has a `ComIPduDirection`. Therefore two `ComIPdu`s (Tx and Rx) need to be created from such a System Description.

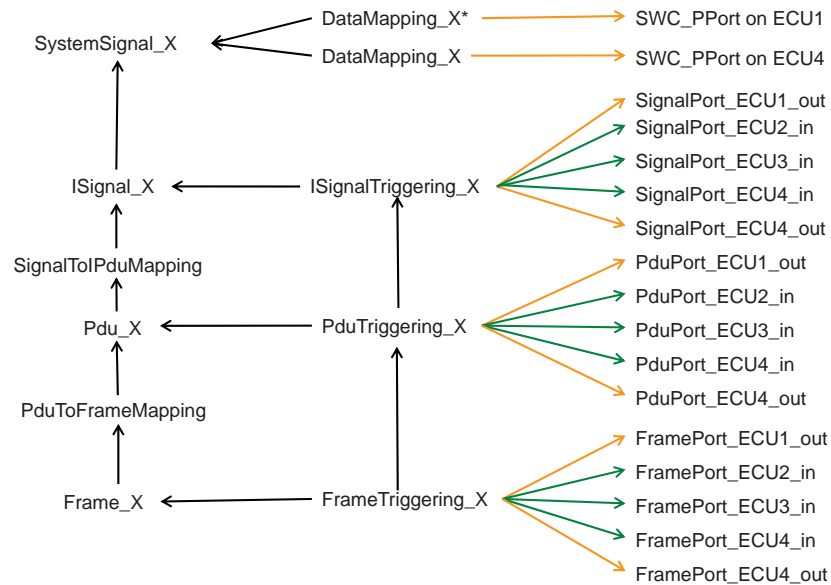


Figure 13.4: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Description with ECU1, ECU2, ECU3 and ECU4)

In case a System Extract / ECU Extract is build, only the relevant `FramePorts`, `IPduPorts` and `ISignalPorts` for the corresponding `EcuInstance` are extracted. Especially in case an additional `EcuInstance` is designed to send and receive the same `Frame` all the other ECU extracts will not be affected by this change. Figure 13.5 shows a System Extract where only the description for ECU4 is available. Please note that in this example the `VariableDataPrototype` in the PPort and the `VariableDataPrototype` in the RPort of the Software Component are mapped to the same `SystemSignal`.

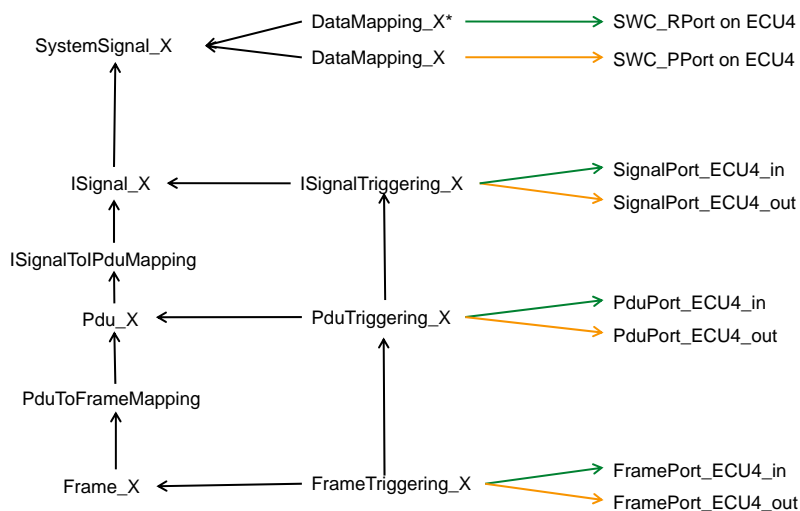


Figure 13.5: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Extract with ECU4 only)

13.3 Support of dynamic CAN IDs

To support efficient diagnostics with on-board clients, efficient routing, and efficient SAE J1939 transport protocol and request handling, AUTOSAR provides access to dynamic CAN `identifier` parts in upper layers of the COM stack. This is achieved by appending parts of the `identifier` (or the complete `identifier`) as `MetaData` to the `Pdu` payload. The usage of `MetaData` is an Ecu Configuration decision. A System Description does not define whether `MetaData` shall be used or not.

The System Template uses the following attributes for the configuration of dynamic CAN IDs:

- The `rxMask` of a `CanFrameTriggering` defines the relevant bits in a CAN `identifier` and thus defines a range of CAN `identifiers` that match these bits and may vary in the other bits.
- The `txMask` of a `CanFrameTriggering` defines the static bits in a CAN `identifier` and thus allows to set the other bits using the data appended to the payload.

These parameters are sufficient to support the following scenarios:

- A `Pdu` is transmitted from one AUTOSAR node to another with variable ID parts. In this case, `rxMask` and `txMask` will be identical, and the variable `identifier` parts placed in the `Pdu MetaData` by the sender will be routed transparently and received in the same way.
- A `Pdu` is transmitted by one node with a static `identifier` and received using the `rxMask`. In this case, the `MetaData` is not used, and the receiver is tolerant regarding dynamic address parts.
- J1939 `Pdu` is sent with fixed priority, but priority is ignored by the receiver. Here, the `MetaData` may or may not be used, and the `rxMask` differs from the `txMask` just in the three priority bits.

13.4 N:1 Sender Receiver communication description in a System Extract over one `PhysicalChannel`

Description: The System Template supports a System Extract description of a n:1 sender-receiver communication over one `PhysicalChannel` where each sender and the receiver are located on different Ecus. Each sender Ecu sends the same data marked with a different frame identifier (e.g. CAN Identifier) to the receiver Ecu over the `PhysicalChannel`.

Implementation: This usage shall be supported by defining one `Frame` and several `FrameTriggerings` on the same `PhysicalChannel`. Each defined `FrameTriggering` refers to the same `Frame`. The senders and receivers of a specific `FrameTriggering` are defined with references to `FramePorts`.

For every defined `Pdu` that is contained in the `Frame` exactly one `PduTriggering` is defined. This also means that all defined `FrameTriggerings` refer to the same `PduTriggerings` with the `FrameTriggering.pduTriggering` reference.

The communication direction of the `Pdu` is defined by `PduTriggering` references to `IPduPorts`. All sender `IPduPorts` and receiver `IPduPorts` are referenced by the same `PduTriggering`.

The description of `ISignals` and `ISignalTriggerings` shall be defined accordingly. Please also note that in case of n:1 sender-receiver communication each sender shall be represented by the same `SystemSignal` according to [constr_3086].

Example: In figure 13.6 a small example is shown. Three different Ecus (Ecu1, Ecu2, Ecu3) are sending the same `Frame` to Ecu4.

- Ecu1 sends the Frame with `CanId` = 3 as described with `FrameTriggering3`.
- Ecu2 sends the Frame with `CanId` = 2 as described with `FrameTriggering2`.
- Ecu3 sends the Frame with `CanId` = 1 as described with `FrameTriggering1`.

The `Frame` contains one single `Pdu`. Only one `PduTriggering` is defined here that refers to three `IPduPorts` with `communicationDirection` "out" (`Ecu1OutPort`, `Ecu2OutPort` and `Ecu3OutPort`) and to one `IPduPort` with `communicationDirection` "in" (`Ecu4InPort`). Please note that the references between the Triggering elements (`FrameTriggering.pduTriggering` and `PduTriggering.iSignalTriggering`) are not visible in figure 13.6 for the sake of clarity.

The description of the `ISignal` that is included in the `Pdu` and the `ISignalTriggering` is defined accordingly.

Upstream Mapping: In the basic Ecu configuration for the receiving Ecu that is derived from such a System Extract all `FrameTriggerings` shall be mapped to the same `Pdu` that is passed to a upper layer module (e.g. `Nm`, `PduR`). This corresponds to the upstream mapping rules for COM Signals defined in [TPS_SYST_01066] and [TPS_SYST_01067].

- `CanIf`: several `CanIfRxPduCfg` containers need to be created with different `CanIfRxPduCanIds` that all point to the same `Pdu` (`CanIfRxPduRef`).
- `Frlf`: several `FrlfFrameTriggering` containers need to be created that all point to the same `Pdu` (`FrlfFrameStructure/FrlfPduInFrame/FrlfPduRef`)

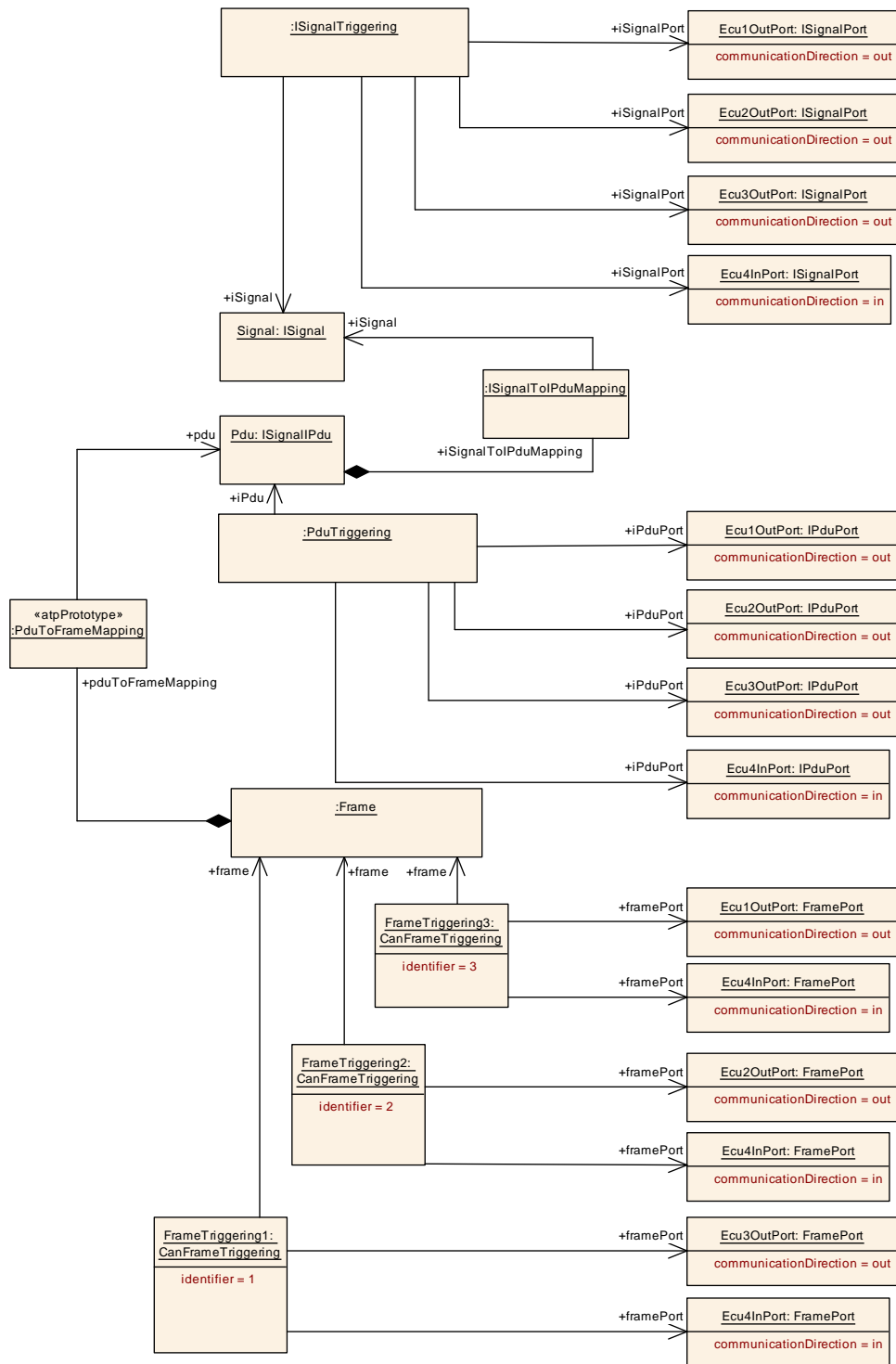


Figure 13.6: Example for a N:1 Sender Receiver communication description in a System Extract

13.5 Description of MOST Functions

The MOST communication protocol is not supported by the AUTOSAR Basic Software but it is possible to convert FIBEX [9] descriptions with MOST content to an AUTOSAR description. This chapter describes how MOST Functions may be described with the means of the Software Component Template [5].

FIBEX supports the description of SW-PACKAGES (represents a bundle of FBlocks and implemented MOST functions), MOST-FUNCTION-BLOCKS (contain functions with operation types and finally parameters, e.g. CD Player), MOST-FUNCTIONS (e.g. a CD player possesses functions such as Play, Stop, Eject, and Time Played) and OP-TYPEs (operations that are applied to the respective function, e.g. Play.Set(tracknumber)). The following table shows how the FIBEX elements may be converted into an AUTOSAR description.

MOST FIBEX Element	Description	AUTOSAR Element	Mapping Rule
FUNCTION-BLOCK	A MOST device contains multiple components that are called function blocks, for example, tuner, amplifier, or CD player.	SwComponentType	Each FunctionBlock shall be described as a SwComponentType
FUNCTION-BLOCK-INSTANCE	There may be several Instances with the same FBlockID in the system (two CD changers, four active speakers, several diagnosis blocks)	SwComponentPrototype	Each FunctionBlockInstance shall be described as a SwComponentPrototype
MOST-FUNCTION	Methods and Properties of a Function Block (e.g. Play, Stop...)	ClientServerInterface	Methods and Properties shall be described as ClientServerInterfaces
OP-TYPE	The OPType indicates which operation shall be applied to the property or method (e.g. Play.Start, Property.Get)	ClientServerOperation	Methods and Properties shall be described as ClientServerOperations .
OP-TYPE Parameter	Parameters of OP-TYPE (e.g. tracknumber)	ArgumentDataPrototype	OP-TYPE Parameters shall be described as ArgumentDataPrototypes of ClientServerOperations .

CLUSTER (MOST-Cluster)	MOST CommunicationCluster	UserDefinedCluster	A MOST CommunicationCluster shall be described as UserDefinedCluster that allows the modeling of arbitrary Communication Clusters. A MOST-Cluster may aggregate several PhysicalChannels
CHANNEL	The CHANNEL object is used to specify the communications channel used by individual OPTypes.	UserDefinedPhysicalChannel	A UserDefinedPhysicalChannel shall be described for each CHANNEL (Control Channel and/or a MOST High Protocol) that is used by the MOST CommunicationCluster.
PDU TRIGGERING	The PDU-TRIGGERING is created for every OP-TYPE that is transported on this CHANNEL.	PduTriggering	A PduTriggering shall be created for every Pdu that contains ClientServerOperations that correspond to a OP-Type and shall be transported on the PhysicalChannel that aggregates this PduTriggering.
PDU	In FIBEX the OP-TYPE corresponds to a PDU in the communication description	Pdu	In AUTOSAR the ClientServerOperation representing the OP-TYPE shall be mapped with the ClientServerToSignalMapping to a SystemSignal. For the SystemSignal an ISignal shall be created. The ISignal is mapped into an ISignalIPdu.

Table 13.1: Mapping of MOST FIBEX elements to AUTOSAR elements

A Glossary

Artifact This is a Work Product Definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts ([39]).

At a high level, an artifact is represented as a single conceptual file.

AUTOSAR Tool This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an authoring tool, a converter tool, a processor tool or as a combination of those (see separate definitions).

AUTOSAR Authoring Tool An AUTOSAR Tool used to create and modify AUTOSAR XML Descriptions. Example: System Description Editor.

AUTOSAR Converter Tool An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files. Example: ECU Flattener

AUTOSAR Definition This is the definition of parameters which can have values. One could say that the parameter values are Instances of the definitions. But in the meta model hierarchy of AUTOSAR, definitions are also instances of the meta model and therefore considered as a description. Examples for AUTOSAR definitions are: `EcucParameterDef`, `PostBuildVariantCriterion`, `SwSystemconst`.

AUTOSAR XML Description In AUTOSAR this means "filled Template". In fact an AUTOSAR XML description is the XML representation of an AUTOSAR model.

The AUTOSAR XML description can consist of several files. Each individual file represents an AUTOSAR partial model and shall validate successfully against the AUTOSAR XML schema.

AUTOSAR Meta-Model This is an UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR meta-model is an UML representation of the AUTOSAR templates. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes, UML tags and OCL expressions (object constraint language) are used for defining specific semantics and constraints.

AUTOSAR Meta-Model Tool The AUTOSAR Meta-Model Tool is the tool that generates different views (class tables, list of constraints, diagrams, XML Schema etc.) on the AUTOSAR meta-model.

AUTOSAR Model This is a representation of an AUTOSAR product. The AUTOSAR model represents aspects suitable to the intended use according to the AUTOSAR methodology.

Strictly speaking, this is an instance of the AUTOSAR meta-model. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR meta-model.

AUTOSAR Partial Model In AUTOSAR, the possible partitioning of models is marked in the meta-model by `<<atpSplittable>>`. One partial model is represented in an AUTOSAR XML description by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.

AUTOSAR Processor Tool An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files. Example: RTE Generator

AUTOSAR Specification Element An AUTOSAR Specification Element is a named element that is part of an AUTOSAR specification. Examples: requirement, constraint, specification item, class or attribute in the meta model, methodology, deliverable, methodology activity, model element, bsw module etc.

AUTOSAR Template The term "Template" is used in AUTOSAR to describe the format different kinds of descriptions. The term template comes from the idea, that AUTOSAR defines a kind of form which shall be filled out in order to describe a model. The filled form is then called the description.

In fact the AUTOSAR templates are now defined as a meta-model.

AUTOSAR Validation Tool A specialized `AUTOSAR Tool` which is able to check an AUTOSAR model against the rules defined by a profile.

AUTOSAR XML Schema This is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR meta-model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.

Blueprint This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta model resp. types, this process is *not* an instantiation.

Instance Generally this is a particular exemplar of a model or of a type.

Life Cycle Life Cycle is the course of development/evolutionary stages of a model element during its life time.

Meta-Model This defines the building blocks of a model. In that sense, a Meta-Model represents the language for building models.

Meta-Data This includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc.

Model A Model is an simplified representation of reality. The model represents the aspects suitable for an intended purpose.

Partial Model This is a part of a model which is intended to be persisted in one particular artifact.

Pattern in GST This is an approach to simplify the definition of the meta model by applying a model transformation. This transformation creates an enhanced model out of an annotated model.

Profile Authoring Support Data Data that is used for efficient authoring of a profile. E.g. list of referable constraints, meta-classes, meta-attributes or other reusable model assets (blueprints)

Profile Authoring Tool A specialized `AUTOSAR Tool` which focuses on the authoring of profiles for data exchange points. It e.g. provides support for the creation of profiles from scratch, modification of existing profiles or composition of existing profiles.

Profile Compatibility Checker Tool A specialized `AUTOSAR Tool` which focuses on checking the compatibility of profiles for data exchange. Note that this compatibility check includes manual compatibility checks by engineers and automated assistance using more formal algorithms.

Profile Consistency Checker Tool A specialized `AUTOSAR Tool` which focuses on checking the consistency of profiles.

Property A property is a structural feature of an object. As an example a “connector” has the properties “receive port” and “send port”

Properties are made variant by the `<<atpVariation>>`.

Prototype This is the implementation of a role of a type within the definition of another type. In other words a type may contain Prototypes that in turn are typed by “Types”. Each one of these prototypes becomes an instance when this type is instantiated.

Type A type provides features that can appear in various roles of this type.

Value This is a particular value assigned to a “Definition”.

Variability Variability of a system is its quality to describe a set of variants. These variants are characterized by variant specific property settings and / or selections. As an example, such a system property selection manifests itself in a particular “receive port” for a connection.

This is implemented using the `<<atpVariation>>`.

Variant A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no variability anymore with respect to the binding time.

This is implemented using `EvaluatedVariantSet`.

Variation Binding A variant is the result of a variation binding process that resolves the variability of the system by assigning particular values/selections to all the system’s properties.

This is implemented by `VariationPoint`.

Variation Binding Time The variation binding time determines the step in the methodology at which the variability given by a set of variable properties is resolved.

This is implemented by `vh.LatestBindingtime` at the related properties.

Variation Definition Time The variation definition time determines the step in the methodology at which the variation points are defined.

Variation Point A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete variant.

This is implemented by `VariationPoint`.

B Detailed Representation of InstanceRef Associations in the System Template

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [2]. This chapter contains the detailed InstanceRef Diagrams.

B.1 Usage of InstanceRefs in Data Mapping diagrams

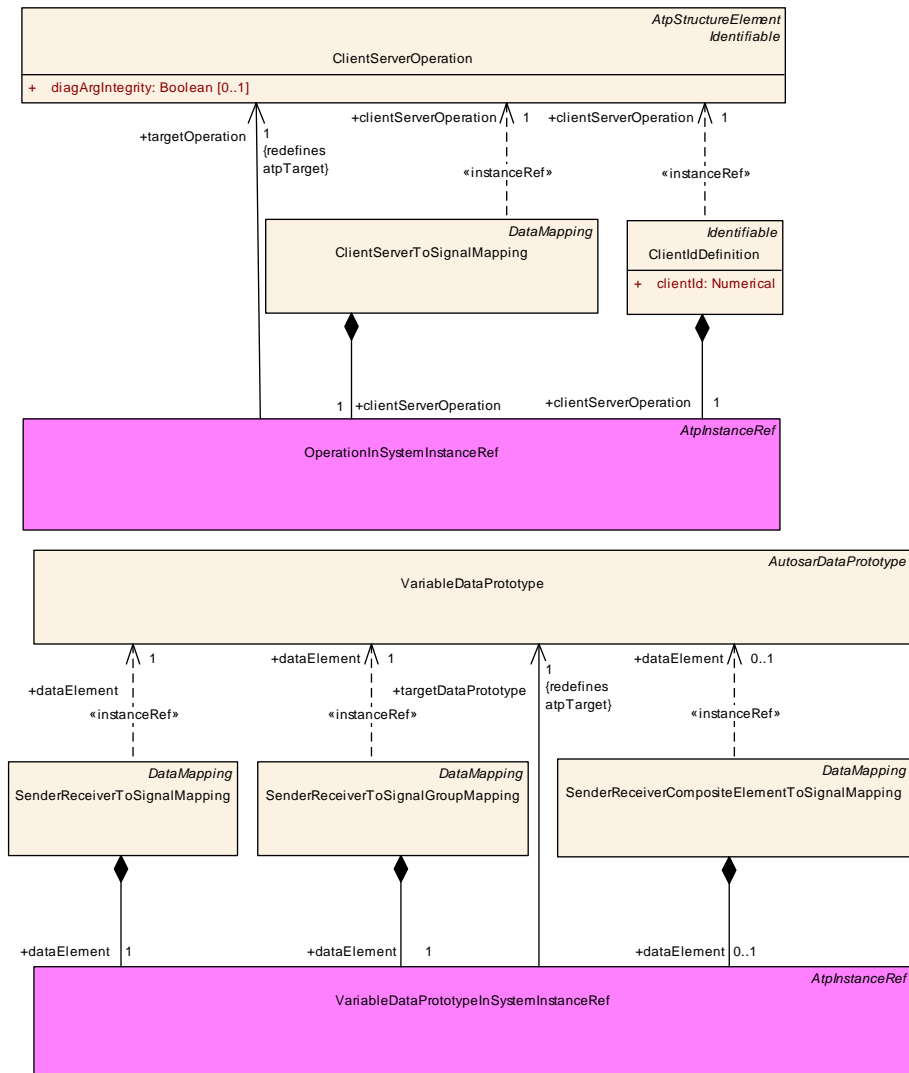


Figure B.1: Data Mapping Instance Ref Usage

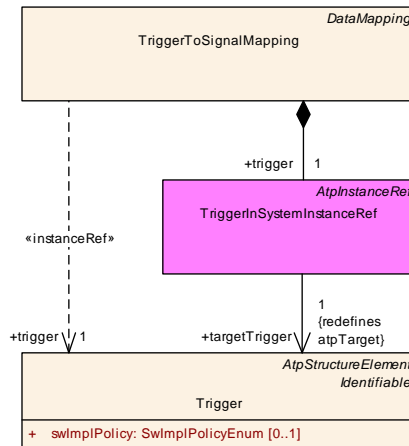


Figure B.2: Modeling of InstanceRef usage for TriggerInSystemInstanceRef

B.2 Usage of InstanceRefs in SW Mapping diagrams

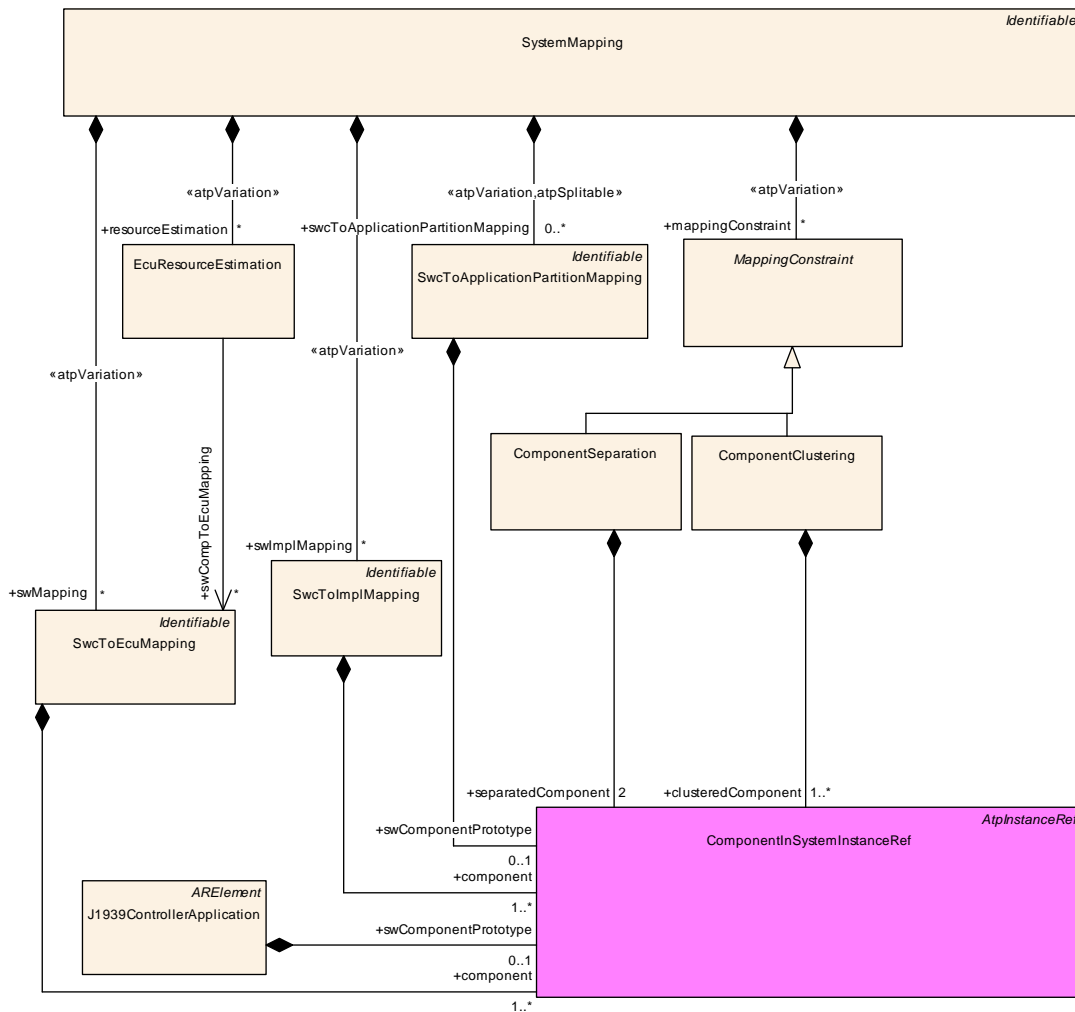


Figure B.3: SW Mapping Instance Ref Usage

B.3 Usage of InstanceRefs in Signal Path Constraint diagrams

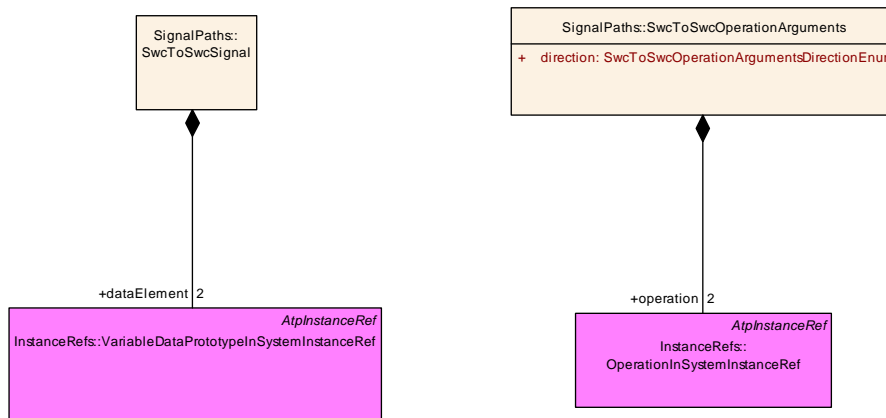


Figure B.4: SW Mapping Instance Ref Usage

B.4 Usage of InstanceRefs in PncMapping

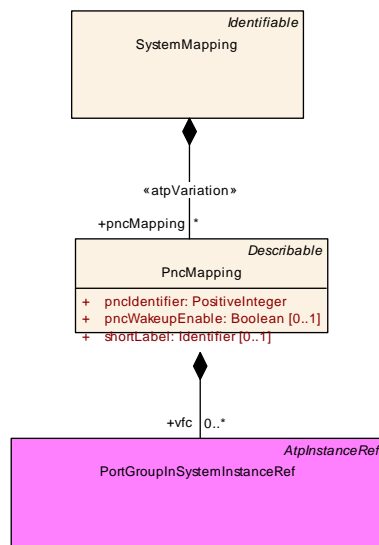


Figure B.5: Partial Network Mapping Instance Ref Usage

B.5 Usage of InstanceRefs in ComManagementMapping

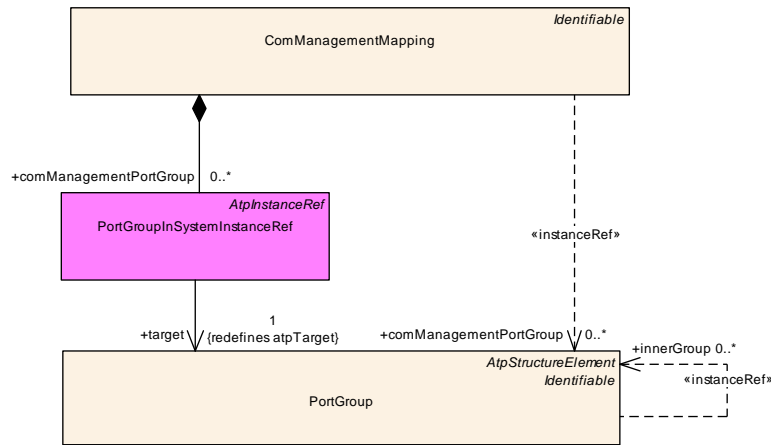


Figure B.6: ComManagementMapping Instance Ref Usage

B.6 "SWC in System" InstanceRef

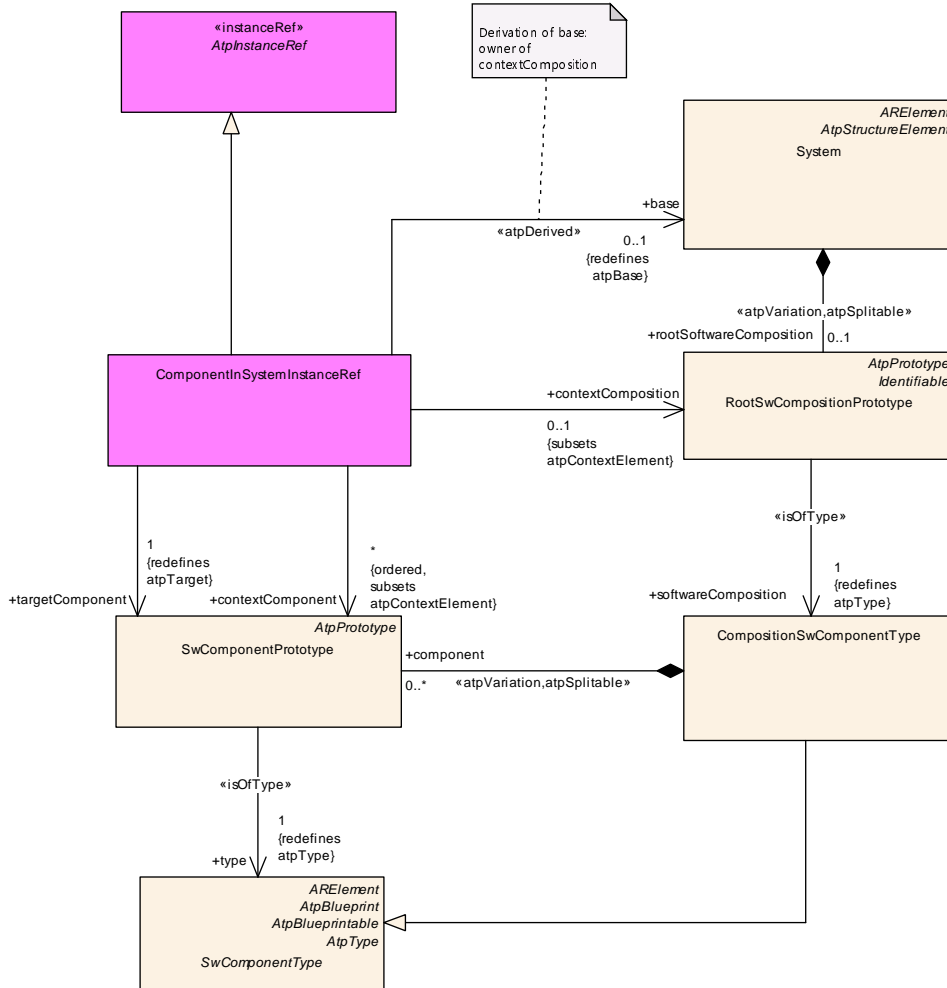


Figure B.7: ComponentInSystem InstanceRef

Class	ComponentInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	Stereotypes: atpDerived Tags: xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	Tags: xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	Tags: xml.sequenceOffset=20





Class	ComponentInSystemInstanceRef			
target Component	SwComponent Prototype	1	ref	Tags:xml.sequenceOffset=40

Table B.1: ComponentInSystemInstanceRef

If the referenced `SwComponentPrototype` is located within the `RootSwCompositionPrototype` of a `System` then the `contextComposition` to the `RootSwCompositionPrototype` shall be provided. In this scenario we have a `System Extract` where the `RootSwComposition` may contain other compositions. If the referenced `SwComponentPrototype` is the `RootSwCompositionPrototype` itself then `contextComposition` reference to the `RootSwCompositionPrototype` shall be skipped and only the `targetComponent` to the `RootSwCompositionPrototype` shall be used. In this scenario we have an `Ecu Extract` where the `RootSwComposition` contains `PortPrototypes` that describe the external communication.

B.7 "Operation in System" InstanceRef

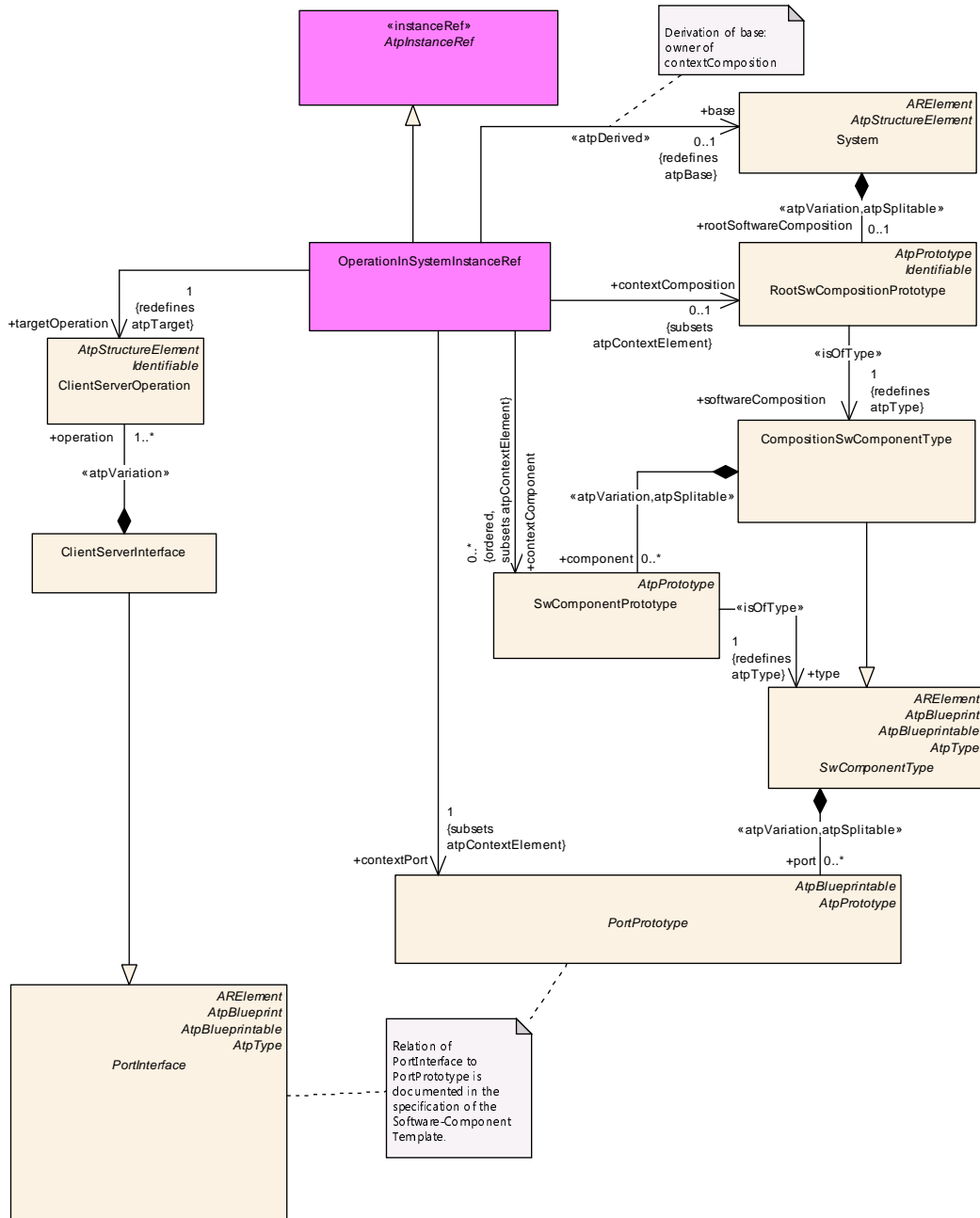


Figure B.8: OperationInSystem InstanceRef

Class	OperationInSystemInstanceRef
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs
Note	
Base	ARObject, AtpInstanceRef





Class	OperationInSystemInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	Stereotypes: atpDerived Tags: xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	Tags: xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	Tags: xml.sequenceOffset=20
contextPort	PortPrototype	1	ref	Tags: xml.sequenceOffset=40
targetOperation	ClientServerOperation	1	ref	Tags: xml.sequenceOffset=50

Table B.2: OperationInSystemInstanceRef

If the referenced `ClientServerOperation` is part of a `PortInterface` of a `SwComponentPrototype` that is located within the `RootSwCompositionPrototype` then the `contextComposition` reference to the `RootSwCompositionPrototype` shall be provided. In this scenario we have a System Extract where the `RootSwComposition` may contain other compositions. If the referenced `ClientServerOperation` is part of a `PortInterface` of the `RootSwCompositionPrototype` itself then the `contextComposition` reference to the `RootSwCompositionPrototype` shall be skipped and the `RootSwCompositionPrototype` shall be referenced as `contextComponent`. In this scenario we have an Ecu Extract where the `RootSwComposition` contains `PortPrototypes` that describe the external communication.

B.8 "VariableDataPrototype" InstanceRef

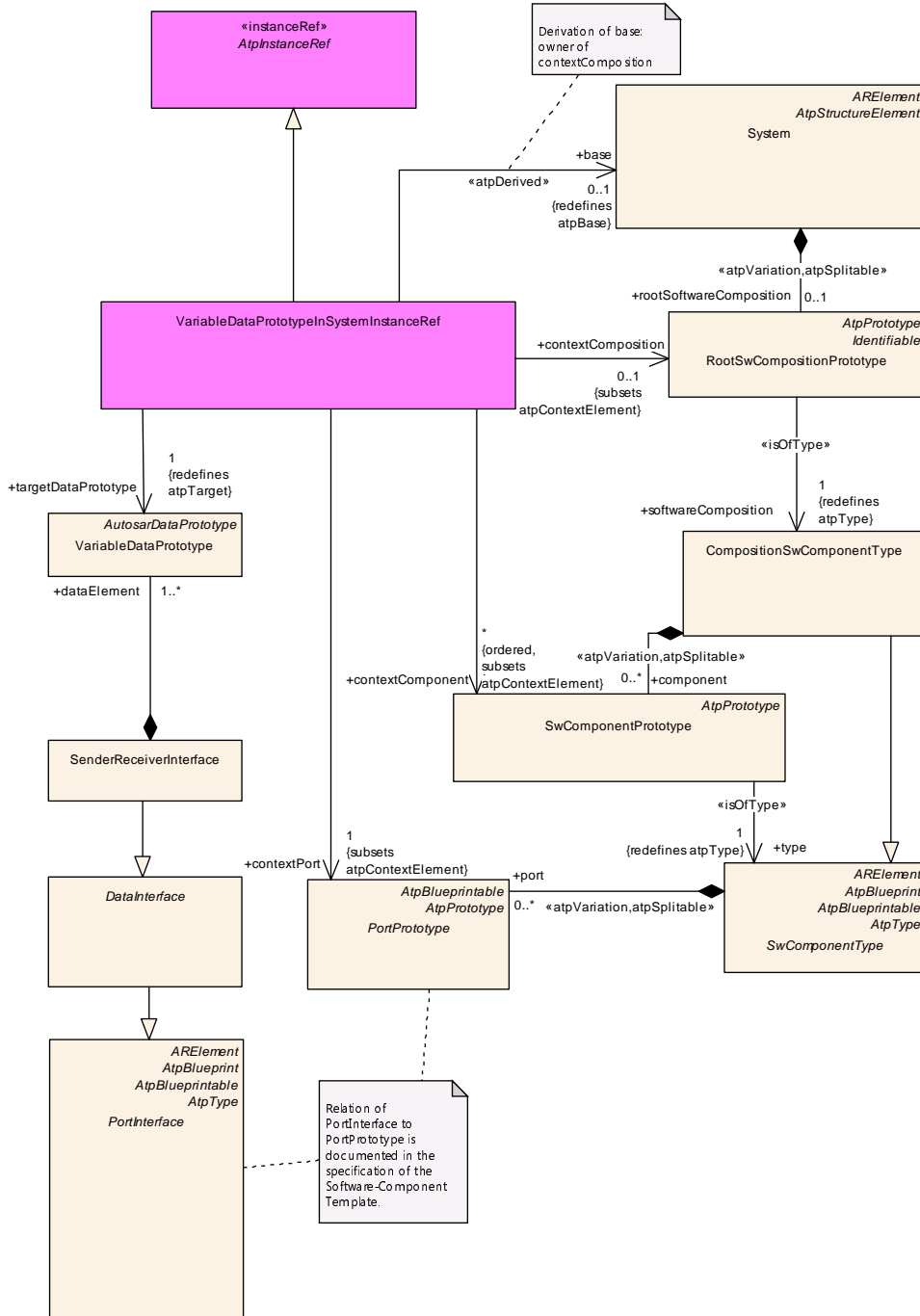


Figure B.9: VariableDataPrototypeInSystem InstanceRef

Class	VariableDataPrototypeInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	<i>ARObject</i> , <i>AtpInstanceRef</i>			
Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	Stereotypes: atpDerived
context Component (ordered)	SwComponent Prototype	*	ref	
context Composition	RootSwComposition Prototype	0..1	ref	
contextPort	PortPrototype	1	ref	
targetData Prototype	VariableDataPrototype	1	ref	

Table B.3: VariableDataPrototypeInSystemInstanceRef

If the referenced [VariableDataPrototype](#) is part of a [PortInterface](#) of a [SwComponentPrototype](#) that is located within the [RootSwCompositionPrototype](#) then the [contextComposition](#) reference to the [RootSwCompositionPrototype](#) shall be provided. In this scenario we have a System Extract where the [RootSwComposition](#) may contain other compositions. If the referenced [VariableDataPrototype](#) is part of a [PortInterface](#) of the [RootSwCompositionPrototype](#) itself then the [contextComposition](#) reference to the [RootSwCompositionPrototype](#) shall be skipped and the [RootSwCompositionPrototype](#) shall be referenced as [contextComponent](#). In this scenario we have an Ecu Extract where the [RootSwComposition](#) contains [PortPrototypes](#) that describe the external communication.

Please note that the `xml.sequenceOffset` is not set for this `InstanceRef` and therefore the properties are serialized in an alphabetical order.

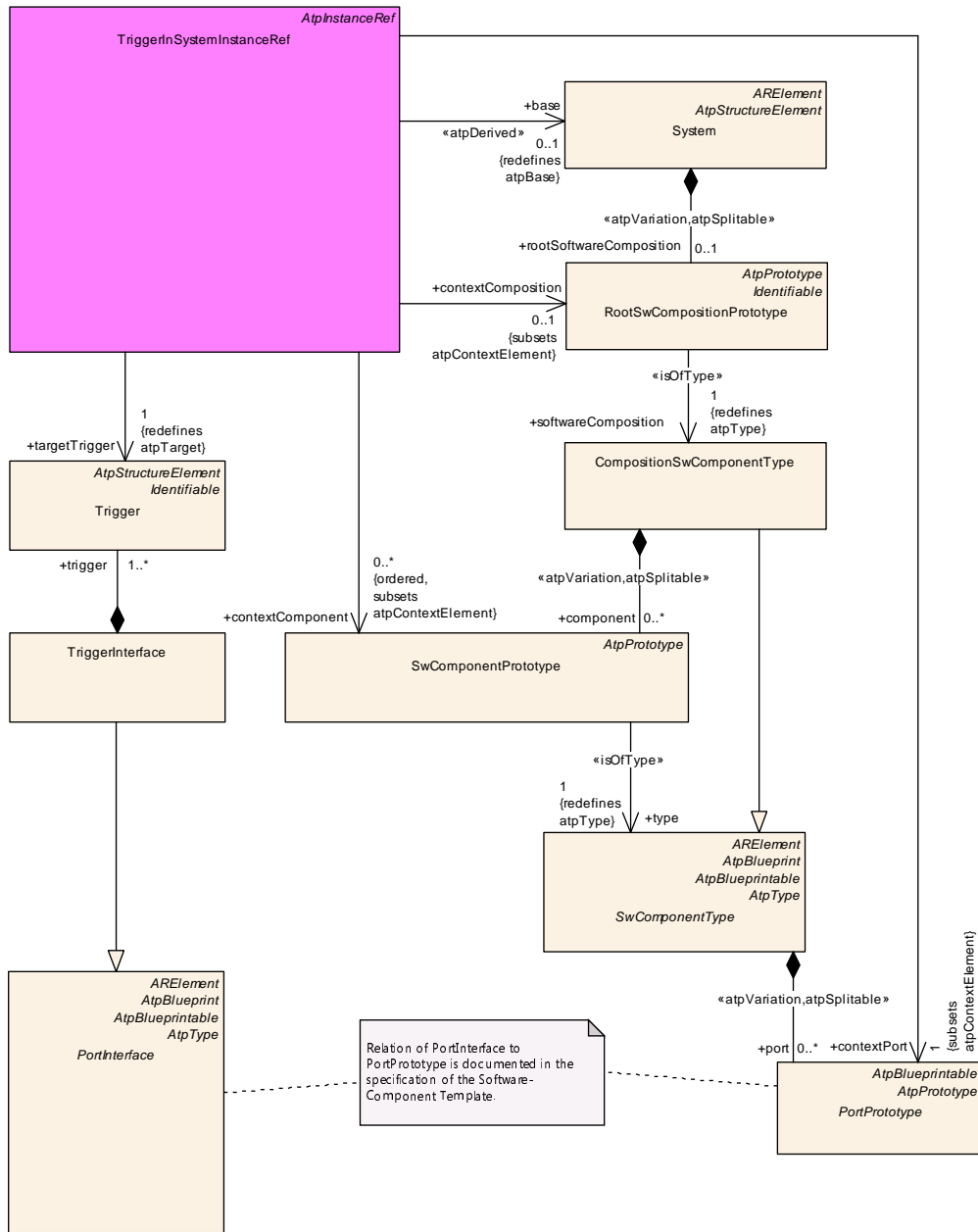


Figure B.10: TriggerInSystemInstanceRef

Class	TriggerInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	This represents that base of the InstanceRef Stereotypes: atpDerived Tags: xml.sequenceOffset=10





Class	TriggerInSystemInstanceRef			
context Component (ordered)	SwComponent Prototype	*	ref	This represents the set of context components. The association is ordered because it needs to respect the nesting order. Tags: xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	This represents the reference to the RootSwComposition type representing a context of the Instance Ref. Tags: xml.sequenceOffset=20
contextPort	PortPrototype	1	ref	This represents the PortPrototype in which the target Trigger is located. Tags: xml.sequenceOffset=40
targetTrigger	Trigger	1	ref	This represents the target Trigger. Tags: xml.sequenceOffset=50

Table B.4: TriggerInSystemInstanceRef

If the referenced [Trigger](#) is part of a [PortInterface](#) of a [SwComponentPrototype](#) that is located within the [RootSwCompositionPrototype](#) then the [base](#) reference and the [contextComposition](#) reference to the [RootSwCompositionPrototype](#) shall be provided. If the referenced [Trigger](#) is part of a [PortInterface](#) of the [RootSwCompositionPrototype](#) itself then the [base](#) reference and the [contextComposition](#) reference to the [RootSwCompositionPrototype](#) shall be skipped and the [RootSwCompositionPrototype](#) shall be referenced as [contextComponent](#).

B.9 "PortGroup in System" InstanceRef

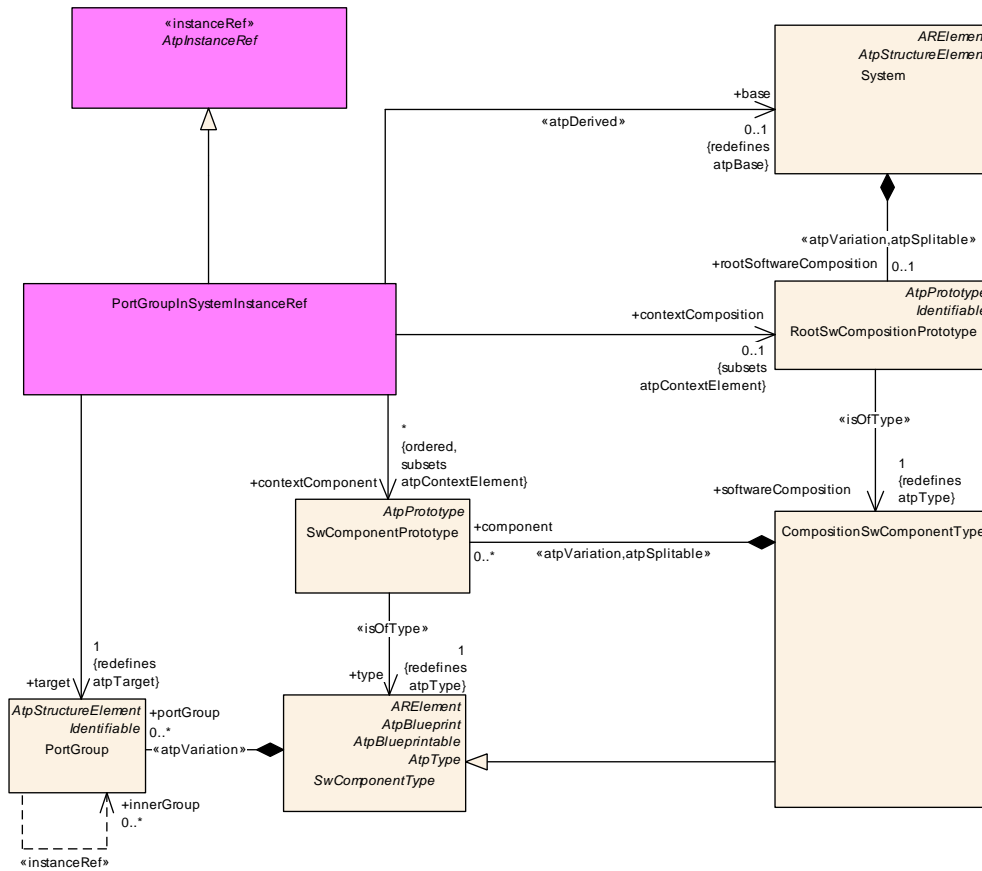


Figure B.11: PortGroupInSystem InstanceRef

Class	PortGroupInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	Stereotypes: atpDerived Tags: xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	Tags: xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	Tags: xml.sequenceOffset=20
target	PortGroup	1	ref	Link to a PortGroup that is defined in a component which is part of this CompositionSwComponentType. Tags: xml.sequenceOffset=40

Table B.5: PortGroupInSystemInstanceRef

If the referenced [PortGroup](#) is part of a [SwComponentPrototype](#) that is located within the [RootSwCompositionPrototype](#) then the [contextComposition](#) reference to the [RootSwCompositionPrototype](#) shall be provided. In this scenario we

have a System Extract where the `RootSwComposition` may contain other compositions. If the referenced `PortGroup` is part of the `RootSwCompositionPrototype` itself then the `contextComposition` reference to the `RootSwCompositionPrototype` shall be skipped and the `RootSwCompositionPrototype` shall be referenced as `contextComponent`. In this scenario we have an Ecu Extract where the `RootSwComposition` contains `PortPrototypes` that describe the external communication.

B.10 "DataPrototype in PortInterface" InstanceRef

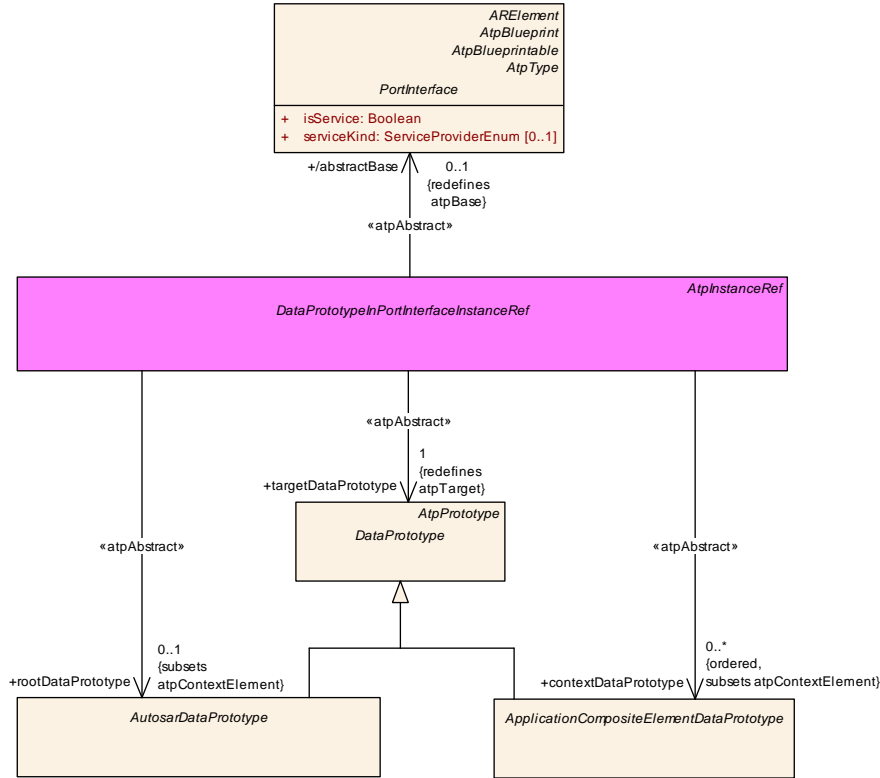


Figure B.12: DataPrototypeInPortInterfaceInstanceRef InstanceRef

Class	<i>DataPrototypeInPortInterfaceInstanceRef</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note	This meta-class represents the ability to: <ul style="list-style-type: none"> refer to a DataPrototype in the context of a PortInterface. refer to the internal structure of a DataPrototype which is typed by an ApplicationDatatype in the context of a PortInterface. 			
Base	<i>ARObject</i> , <i>AtpInstanceRef</i>			
Subclasses	DataPrototypeInClientServerInterfaceInstanceRef , DataPrototypeInSenderReceiverInterfaceInstanceRef			
Attribute	Type	Mult.	Kind	Note
abstractBase	PortInterface	0..1	ref	Stereotypes: atpAbstract
contextData Prototype (ordered)	ApplicationCompositeElementDataPrototype	*	ref	Stereotypes: atpAbstract Tags: xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	Stereotypes: atpAbstract Tags: xml.sequenceOffset=10
targetData Prototype	DataPrototype	1	ref	Stereotypes: atpAbstract Tags: xml.sequenceOffset=30

Table B.6: DataPrototypeInPortInterfaceInstanceRef

If the referenced target [DataPrototype](#) is the root [AutosarDataPrototype](#) in a [PortInterface](#) then only the [targetDataPrototype](#) reference shall be provided.

If the referenced `DataPrototype` is part of a root `AutosarDataPrototype` that is part of a `PortInterface` then the `rootDataPrototype` shall be provided. The referenced `ApplicationCompositeElementDataPrototype` can be arbitrarily nested within a `DataPrototype`. In such a case additional `contextDataPrototype` references shall be provided.

Please note that the specializations `DataPrototypeInSenderReceiverInterfaceInstanceRef` and `DataPrototypeInClientServerInterfaceInstanceRef` work in the same way.

C Harmonisation between Upstream Templates and ECU Configuration

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template and ECU Resource Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

- How shall a supplier use the information in a System Description in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an ECU Configuration Description out of ECU Extract of System Description?

In addition to adhering to the mapping rules defined in this appendix an automated generation of an ECU Configuration Description out of ECU Extract of System Description should apply a certain implementation-specific name mangling when deriving the `shortName` of the `EcucContainerValue` elements to ensure that the resulting ECU Configuration Description is valid with respect to `constr_2508` of [2].

Please note that the tables contain the following columns:

bsw module: Name of BSW module

bsw context: Reference to parameter container

bsw type: Type of parameter

bsw param: Name of the BSW parameter

bsw desc: Description from the configuration document

m2 template: System Template, SW Component Template, ECU Resource Template

m2 param: Name of the upstream template parameter

m2 description: Description from the upstream template definition

mapping rule: Textual description on how to transform between M2 and BSW domains

mapping type:

- local: no mapping needed since parameter local to BSW
- partial: some data can be automatically mapped but not all
- full: all data can be automatically mapped

C.1 ComStack

C.1.1 Com Mapping

BSW Module	BSW Context	
Com	Com	
BSW Parameter		BSW Type
ComConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR COM module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00337]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		BSW Type
ComDataMemSize		EcucIntegerParamDef
BSW Description		
Size of internal Com data in units of bytes (static memory allocation) - memory required by post-build configuration must be smaller than this constant. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00783]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		BSW Type
ComGwMapping		EcucParamConfContainerDef
BSW Description		
Each instance of this container defines one mapping of the integrated Signal Gateway.		
Template Description		
Arranges those signals (or SignalGroups) that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a source and a target referencing to a ISignalTriggering.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::ISignalMapping		
Mapping Rule		Mapping Type

<p>In the System Extract an explicit ISignalMapping or an implicit ISignalMapping may be defined.</p> <p>Explicit Mapping: Create Container for each ISignalMapping.sourceSignal where the referenced ISignalTriggering refers to an ISignal.</p> <p>Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignal Triggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00544]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping	
BSW Parameter	BSW Type	
ComGwDestination	EcucChoiceContainerDef	
BSW Description		
Each instance of this choice container allows to define one routing destination either by reference to an already configured COM signal / group signal or by a destination description container.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::ISignalMapping.targetSignal		
Mapping Rule		Mapping Type
<p>Explicit Mapping: Create Container for each targetSignal reference that is defined in the ISignalMapping.</p> <p>Implicit Mapping: If the ISignalMapping.targetSignal refers to an ISignal Triggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the target ISignal matches the shortName of a source ISignal of the ISignal Mapping.sourceSignal ISignalGroup.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00546]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination	
BSW Parameter	BSW Type	
ComGwDestinationDescription	EcucParamConfContainerDef	
BSW Description		
Description of a gateway destination. This container allows defining a gateway destination without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		

Mapping Rule		Mapping Type
Informations can be derived from ISignalToIPduMapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00549]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		BSW Type
ComBitPosition		EcucIntegerParamDef
BSW 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		
Template Description		
<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>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00259]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		BSW Type
ComFilter		EcucParamConfContainerDef
BSW Description		
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.		
Template Description		
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.		
M2 Parameter		
CommonStructure::Filter::DataFilter		
Mapping Rule		Mapping Type
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00339]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterAlgorithm		EcucEnumerationParamDef
BSW Description		
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.		
Template Description		
This attribute specifies the type of the filter.		
M2 Parameter		
CommonStructure::Filter::DataFilter.dataFilterType		
Mapping Rule		Mapping Type
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00146]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterMask		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Mask for old and new value.		
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00235]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterMax		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the upper boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00317]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterMin		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the lower boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00318]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterOffset		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Range = 0..(ComFilterPeriod-1)		
Template Description		
Specifies the initial number of messages to occur before the first message is passed		
M2 Parameter		
CommonStructure::Filter::DataFilter.offset		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00313]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterPeriod		EcucIntegerParamDef
BSW Description		
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.		
Template Description		
Specifies number of messages to occur before the message is passed again		
M2 Parameter		
CommonStructure::Filter::DataFilter.period		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00312]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter	
BSW Parameter		BSW Type
ComFilterX		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to compare with		
M2 Parameter		
CommonStructure::Filter::DataFilter.x		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00147]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription	
BSW Parameter		BSW Type
ComGwIPduRef		EcucReferenceDef
BSW Description		
Reference to an I-PDU of a Signal Gateway source or destination description.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00550]

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription	
BSW Parameter		BSW Type
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
Template Description		
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.		
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByte Order	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00157]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description
BSW Parameter	BSW Type
ComSignalInitValue	EcucStringParamDef
BSW 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.	
Template Description	
Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue	
Mapping Rule	Mapping Type
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00170]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description
BSW Parameter	BSW Type
ComTransferProperty	EcucEnumerationParamDef
BSW 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.	
Template Description	
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.	
M2 Parameter	

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00232]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description
BSW Parameter	BSW Type
ComUpdateBitPosition	EcucIntegerParamDef
BSW Description	
<p>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.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>	
Template Description	
<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>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndicationBitPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00257]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination
BSW Parameter	BSW Type
ComGwSignal	EcucParamConfContainerDef
BSW Description	
This container allows specifying a gateway source or destination respectively with a reference to a ComSignal or a ComGroupSignal.	

Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalTriggering.iSignal	
Mapping Rule	Mapping Type
<p>Explicit Mapping: Create Container if ISignal is referenced from ISignalMapping.sourceSignal or ISignalMapping.targetSignal via ISignalTriggering.</p> <p>Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the source ISignal matches the short Name of a destination ISignal of the ISignalMapping.targetSignal ISignalGroup. If the ISignalMapping.targetSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the target ISignal matches the short Name of a source ISignal of the ISignalMapping.sourceSignal ISignalGroup.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00551]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwSignal
BSW Parameter	BSW Type
ComGwSignalRef	EcucChoiceReferenceDef
BSW Description	
Reference to an object of a gateway relation. Either to a ComSignal or a ComGroupSignal.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	Mapping Type
Refers to the to be routed ComSignal or ComGroupSignal.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00547]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping
BSW Parameter	BSW Type
ComGwSource	EcucChoiceContainerDef
BSW Description	
This choice container allows the definition of the gateway source signal either by reference to an already configured COM signal / group signal or by a source description container.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Multiplatform::ISignalMapping.sourceSignal	
Mapping Rule	Mapping Type

<p>Explicit Mapping: Create Container for sourceSignal reference that is defined in the ISignalMapping.</p> <p>Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignal Triggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00545]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource	
BSW Parameter	BSW Type	
ComGwSignal	EcucParamConfContainerDef	
BSW Description		
This container allows specifying a gateway source or destination respectively with a reference to a ComSignal or a ComGroupSignal.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalTriggering.iSignal		
Mapping Rule	Mapping Type	
<p>Explicit Mapping: Create Container if ISignal is referenced from ISignalMapping.sourceSignal or ISignalMapping.targetSignal via ISignalTriggering.</p> <p>Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the source ISignal matches the short Name of a destination ISignal of the ISignalMapping.targetSignal ISignalGroup. If the ISignalMapping.targetSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignalGroup where the shortName of the target ISignal matches the short Name of a source ISignal of the ISignalMapping.sourceSignal ISignalGroup.</p>	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00551]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSignal	
BSW Parameter	BSW Type	
ComGwSignalRef	EcucChoiceReferenceDef	
BSW Description		
Reference to an object of a gateway relation. Either to a ComSignal or a ComGroupSignal.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		

Mapping Rule		Mapping Type
Refers to the to be routed ComSignal or ComGroupSignal.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00547]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource	
BSW Parameter		BSW Type
ComGwSourceDescription		EcucParamConfContainerDef
BSW Description		
Description of a gateway source. This container allows defining a gateway source without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type
Informations can be derived from ISignalToIPduMapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00548]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		BSW Type
ComBitPosition		EcucIntegerParamDef
BSW 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		
Template Description		
<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>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00259]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	

BSW Parameter		BSW Type	
ComBitSize		EcucIntegerParamDef	
BSW 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.			
Template Description			
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.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Com_00158]	

BSW Module		BSW Context	
Com		Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		BSW Type	
ComGwIPduRef		EcucReferenceDef	
BSW Description			
Reference to an I-PDU of a Signal Gateway source or destination description.			
Template Description			
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping			
Mapping Rule		Mapping Type	
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Com_00550]	

BSW Module		BSW Context	
Com		Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		BSW Type	
ComSignalEndianness		EcucEnumerationParamDef	
BSW Description			
Defines the endianness of the signal's network representation.			
Template Description			
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.			
M2 Parameter			

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByte Order	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00157]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	BSW Type
ComSignalLength	EcucIntegerParamDef
BSW Description	
<p>Description:</p> <p>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>	
Template Description	
<p>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.</p> <p>Indicates maximum size for dynamic length signals.</p> <p>The ISignal length of zero bits is allowed.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length	
Mapping Rule	Mapping Type
ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00437]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	BSW Type
ComSignalType	EcucEnumerationParamDef
BSW Description	
<p>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.</p> <p>This type could also be used to reserved appropriate storage in AUTOSAR COM.</p>	
Template Description	

<p>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.</p> <p>If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used.</p> <p>If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps.</p> <p>In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.dataTypePolicy	
Mapping Rule	Mapping Type
<p>The mapping depends from the setting in the ISignal.dataTypePolicy:</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationDataType.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_02079 and access SwDataDefProps.swBaseType. <p>Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => ComSignalType should not be configured.</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if SystemSignal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true 	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00127]

BSW Module	BSW Context
-------------------	--------------------

Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		BSW Type
ComUpdateBitPosition		EcucIntegerParamDef
BSW Description		
<p>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.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>		
Template Description		
<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>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndicationBitPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00257]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		BSW Type
ComIPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
Template Description		
<p>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.</p> <p>A maximum of one dynamic length signal per IPdu is allowed.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu		
Mapping Rule		Mapping Type
create container for each SignalIPdu that is transmitted by the regarded ECU.		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00340]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduCallout	EcucFunctionNameDef	
BSW 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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00387]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduCancellationSupport	EcucBooleanParamDef	
BSW 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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00709]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduCounter	EcucParamConfContainerDef	
BSW Description		
This optional container contains the configuration parameters of PDU Counter.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.pduCounter		
Mapping Rule		Mapping Type
If pduCounter is aggregated by ISignalIPdu then create this container		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00592]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter	BSW Type	
ComIPduCounterErrorNotification	EcucFunctionNameDef	
BSW Description		
Name of Com_CbkCounterErr callback function to be called. If this parameter is omitted no I-PDU counter mismatch notification shall take place.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00003]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter	BSW Type	
ComIPduCounterSize	EcucIntegerParamDef	
BSW Description		
Size of PDU Counter expressed in bits		
Template Description		
Size of PduCounter expressed in bits. Range: 1..8		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterSize		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00593]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter	BSW Type	
ComIPduCounterStartPosition	EcucIntegerParamDef	
BSW Description		
Position of PDU counter expressed in bits from start position of data content of I-PDU (SDU). Note that PDU counter is not allowed to cross a byte border. The parameter ComIPduCounterStartPosition shall define the bit0 of the first byte like in little endian byte order.		
Template Description		
Position of PduCounter expressed in bits. Note that PduCounter is not allowed to cross a byte border.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterStart Position		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00594]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter	BSW Type	
ComIPduCounterThreshold	EcucIntegerParamDef	
BSW Description		
Threshold value of I-PDU counter algorithm, see ECUC_Com_00590.		
Template Description		
Threshold value of IPduCounter algorithm. See AUTOSAR COM Spec for more details.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterThreshold		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00595]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduDirection	EcucEnumerationParamDef	
BSW Description		
The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
Template Description		
Communication Direction of the Connector Port (input or output Port).		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommConnectorPort.communicationDirection		
Mapping Rule		Mapping Type
Find IPduTriggering of the regarded SignalIPdu. The IPduTriggering contains a reference to an IPduPort that is aggregated by the regarded ECU. If the communicationDirection of the CommConnectorPort is "in" than the IPdu is received.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00493]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduGroupRef	EcucReferenceDef	
BSW Description		
Reference to the I-PDU groups this I-PDU belongs to.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPduGroup.iSignalIPdu		
Mapping Rule		Mapping Type
Find IPduGroup that points to this SignalIPdu and create the reference.		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00206]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduHandleId	EcucIntegerParamDef	
BSW Description		
<p>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.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00175]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduMainFunctionRef	EcucChoiceReferenceDef	
BSW Description		
<p>Reference to the Com_MainFunctionRx/Com_MainFunctionTx this I-PDU belongs to.</p> <p>Mandatory, if multiple main functions of the relevant type are defined.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10012]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduReplication	EcucParamConfContainerDef	
BSW Description		
<p>This optional container contains the information needed for each I-PDU replicated.</p>		
Template Description		
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.pduReplication	
Mapping Rule	Mapping Type
If pduReplication is defined for the SignalIPdu then create this container	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00599]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduReplication	
BSW Parameter		BSW Type
ComIPduReplicaRef		EcucReferenceDef
BSW Description		
Reference to replicas PduR PDUs of this IPDU.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduReplication.replicaPdus		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00601]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduReplication	
BSW Parameter		BSW Type
ComIPduReplicationQuorum		EcucIntegerParamDef
BSW Description		
The number of identical I-PDUs needed for successful voting.		
Template Description		
Number of identical IPdus needed for successful voting (1-3).		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduReplication.pduReplication Voting		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00600]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		BSW Type
ComIPduSignalGroupRef		EcucReferenceDef
BSW Description		
References to all signal groups contained in this I-Pdu		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type

Find ISignal in the ISignalIPdu that refers to a ISignalGroup and create reference to this Group		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00519]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduSignalProcessing	EcucEnumerationParamDef	
BSW Description		
For the definition of the two modes Immediate and Deferred.		
Template Description		
Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduPort.iPduSignalProcessing		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00119]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduSignalRef	EcucReferenceDef	
BSW Description		
References to all signals contained in this I-PDU.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type
Find ISignal in the IPdu which refers to a SystemSignal and create reference to this Signal.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00518]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduTriggerTransmitCallout	EcucFunctionNameDef	
BSW Description		
If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00765]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComIPduType	EcucEnumerationParamDef	
BSW 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.	
Template Description	Contains all configuration elements for AUTOSAR TP.	
M2 Parameter	SystemTemplate::TransportProtocols::TpConfig	
Mapping Rule	Mapping Type	
If this PduTriggering is referenced by a TpConnection then set this Enumeration Literal to TP.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00761]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComPduldRef	EcucReferenceDef	
BSW Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00711]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter	BSW Type	
ComTxIPdu	EcucParamConfContainerDef	
BSW Description	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.	
Template Description	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.	
M2 Parameter	SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu	

Mapping Rule		Mapping Type
create container if an ISignalIPdu is transmitted by the regarded ECU.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00496]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter	BSW Type	
ComMetaDataDefault	EcucStringParamDef	
BSW Description		
<p>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_TriggerIPDUWithMetaData), the AUTOSAR COM module uses this configured default meta data for sending.</p> <p>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).</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10005]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter	BSW Type	
ComMinimumDelayTime	EcucFloatParamDef	
BSW Description		
<p>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.</p>		
Template Description		
Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduTiming.minimumDelay		
Mapping Rule		Mapping Type
Find IPduTiming for the transmitted IPdu and use the specified value.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00181]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter	BSW Type	
ComTxIPduClearUpdateBit	EcucEnumerationParamDef	
BSW Description		

Defines when the update-bits of signals or signal groups, contained in this I-PDU, will be cleared.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00576]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	BSW Type
ComTxIPduUnusedAreasDefault	EcucIntegerParamDef
BSW 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.	
Template Description	
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.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.unusedBitPattern	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00017]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	BSW Type
ComTxModeFalse	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to false.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeDeclaration.transmissionModeFalseTiming	
Mapping Rule	Mapping Type
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00454]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse
BSW Parameter	BSW Type
ComTxMode	EcucParamConfContainerDef

BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming.	
COM supports the following Transmission Modes:	
* Periodic (Cyclic Timing)	
* Direct /n-times (EventControlledTiming)	
* Mixed (Cyclic and EventControlledTiming are assigned)	
* None (no timing is assigned)	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming	
Mapping Rule	Mapping Type
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00351]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	BSW Type
ComTxModeMode	EcucEnumerationParamDef
BSW Description	
The available transmission modes described in [18] shall be extended by the additional mode None.	
The transmission mode None shall not have any further sub-attributes in the ComTxMode object.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming.	
COM supports the following Transmission Modes:	
* Periodic (Cyclic Timing)	
* Direct /n-times (EventControlledTiming)	
* Mixed (Cyclic and EventControlledTiming are assigned)	
* None (no timing is assigned)	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming	
Mapping Rule	Mapping Type
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlledTiming. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00137]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	BSW Type
ComTxModeNumberOfRepetitions	EcucIntegerParamDef

BSW Description	
Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.	
Template Description	
Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.numberOfRepetitions	
Mapping Rule	Mapping Type
ComTxModeNumberOfRepetitions = EventControlledTiming.numberOfRepetitions	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00281]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	BSW Type
ComTxModeRepetitionPeriod	EcucFloatParamDef
BSW Description	
Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.repetitionPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00282]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	BSW Type
ComTxModeTimeOffset	EcucFloatParamDef
BSW Description	
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupStart and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timeOffset	
Mapping Rule	Mapping Type
The value for the True and the False Transmission Mode can be derived from IPduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_Com_00180]
-------	------------------

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	BSW Type
ComTxModeTimePeriod	EcucFloatParamDef
BSW Description	
Defines the repetition period in seconds of the periodic transmission requests in case ComTxMode-Mode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timePeriod	
Mapping Rule	Mapping Type
The value for the True and the False Transmission Mode can be derived from I PduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00178]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	BSW Type
ComTxModeTrue	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to true.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeDeclara-tion.transmissionModeTrueTiming	
Mapping Rule	Mapping Type
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00455]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue
BSW Parameter	BSW Type
ComTxMode	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.	
Template Description	

If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming.

COM supports the following Transmission Modes:

- * Periodic (Cyclic Timing)
- * Direct /n-times (EventControlledTiming)
- * Mixed (Cyclic and EventControlledTiming are assigned)
- * None (no timing is assigned)

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming	
Mapping Rule	Mapping Type
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00351]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	BSW Type
ComTxModeMode	EcucEnumerationParamDef
BSW Description	
The available transmission modes described in [18] shall be extended by the additional mode None.	
The transmission mode None shall not have any further sub-attributes in the ComTxMode object.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming.	
COM supports the following Transmission Modes:	
* Periodic (Cyclic Timing)	
* Direct /n-times (EventControlledTiming)	
* Mixed (Cyclic and EventControlledTiming are assigned)	
* None (no timing is assigned)	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming	
Mapping Rule	Mapping Type
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlledTiming. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00137]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	BSW Type
ComTxModeNumberOfRepetitions	EcucIntegerParamDef
BSW Description	
Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.	
Template Description	

Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.numberOfRepetitions	
Mapping Rule	Mapping Type
ComTxModeNumberOfRepetitions = EventControlledTiming.numberOfRepetitions	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00281]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	BSW Type
ComTxModeRepetitionPeriod	EcucFloatParamDef
BSW Description	
Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.repetitionPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00282]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	BSW Type
ComTxModeTimeOffset	EcucFloatParamDef
BSW Description	
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupStart and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timeOffset	
Mapping Rule	Mapping Type
The value for the True and the False Transmission Mode can be derived from IPduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00180]

BSW Module	BSW Context
-------------------	--------------------

Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode	
BSW Parameter		BSW Type
ComTxModeTimePeriod		EcucFloatParamDef
BSW Description		
Defines the repetition period in seconds of the periodic transmission requests in case ComTxMode-Mode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timePeriod		
Mapping Rule		Mapping Type
The value for the True and the False Transmission Mode can be derived from I PduTiming.TransmissionModeDeclaration.TransmissionModeTiming element		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00178]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		BSW Type
ComIPduGroup		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters of the AUTOSAR COM module's I-PDU groups.		
Template Description		
The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignallPduGroup contains either ISignallPdus or ISignallPduGroups.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignallPduGroup		
Mapping Rule		Mapping Type
Create container for each CoreCommunication::ISignallPduGroup that is contained in the ECU Extract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00341]

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPduGroup	
BSW Parameter		BSW Type
ComIPduGroupGroupRef		EcucReferenceDef
BSW Description		
References to all I-PDU groups that includes this I-PDU group. If this reference is omitted this I-PDU group does not belong to another I-PDU group.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignallPduGroup.containedISignallPduGroup		
Mapping Rule		Mapping Type
If the IPduGroup has a reference to a contained IPduGroup then create this reference. Please note that in COM the contained IPduGroup points to the containing IPduGroup and in System Template the containing ISignallPduGroup points to the contained ISignallPduGroup.		full
Mapping Status		ECUC Parameter ID

valid	[ECUC_Com_00185]
-------	------------------

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPduGroup	
BSW Parameter	BSW Type	
ComIPduGroupHandleId	EcucIntegerParamDef	
BSW Description		
<p>The numerical value used as the ID of this I-PDU Group . The ComIPduGroupHandleId is required by the API calls to start and stop I-PDU Groups.</p> <p>Range: 0 .. (ComSupportedIPduGroups-1)</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00184]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter	BSW Type	
ComMainFunctionRouteSignals	EcucParamConfContainerDef	
BSW Description		
Each element of this container defines one instance of Com_MainFunctionRouteSignals.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_10013]

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionRouteSignals	
BSW Parameter	BSW Type	
ComMainRouteSignalsPartitionRef	EcucReferenceDef	
BSW Description		
Reference to EcucPartition, where the according Com_MainFunction instance is assigned to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_10015]

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionRouteSignals	
BSW Parameter	BSW Type	
ComMainRouteSignalsTimeBase	EcucFloatParamDef	
BSW Description		
<p>The period between successive calls to according instance of Com_MainFunctionRouteSignals in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionRouteSignals is scheduled according to the value configured here.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10016]	

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter	BSW Type	
ComMainFunctionRx	EcucParamConfContainerDef	
BSW Description		
Each element of this container defines one instance of Com_MainFunctionRx.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10011]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionRx	
BSW Parameter	BSW Type	
ComMainRxPartitionRef	EcucReferenceDef	
BSW Description		
Reference to EcucPartition, where the according Com_MainFunction instance is assigned to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10017]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionRx	
BSW Parameter	BSW Type	
ComMainRxTimeBase	EcucFloatParamDef	
BSW Description		
<p>The period between successive calls to according instance of Com_MainFunctionRx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionRx is scheduled according to the value configured here.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_10018]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter	BSW Type	
ComMainFunctionTx	EcucParamConfContainerDef	
BSW Description		
Each element of this container defines one instance of Com_MainFunctionTx.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_10014]

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionTx	
BSW Parameter	BSW Type	
ComMainTxPartitionRef	EcucReferenceDef	
BSW Description		
Reference to EcucPartition, where the according Com_MainFunction instance is assigned to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_10019]

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionTx	
BSW Parameter	BSW Type	
ComMainTxTimeBase	EcucFloatParamDef	
BSW Description		
<p>The period between successive calls to according instance of Com_MainFunctionTx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionTx is scheduled according to the value configured here.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10010]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComMainFunctionTx	
BSW Parameter	BSW Type	
ComPreparationNotification	EcucFunctionNameDef	
BSW Description		
<p>This callback function indicates that the signals/signal groups to be sent via a dedicated Com_MainFunctionTx instance will now be prepared for transmission.</p> <p>If this parameter is omitted no notification shall take place.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10020]	

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter	BSW Type	
ComMaxIPduCnt	EcucIntegerParamDef	
BSW Description		
<p>Maximum number of IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00782]

BSW Module	BSW Context
Com	Com/ComConfig
BSW Parameter	BSW Type
ComSignal	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's signals.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	Mapping Type
<p>A ComSignal container shall be created for an ISignal that is contained in an ISignalIPdu which the Com module is sending.</p> <p>The creation of a ComSignal container may be omitted for an ISignal that is contained in an ISignalIPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00344]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComBitPosition	EcucIntegerParamDef
BSW 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	
Template Description	
<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>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00259]

BSW Module	BSW Context
------------	-------------

Com	Com/ComConfig/ComSignal	
BSW Parameter		BSW Type
ComBitSize	EcucIntegerParamDef	
BSW 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.		
Template Description		
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.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00158]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		BSW Type
ComDataInvalidAction	EcucEnumerationParamDef	
BSW 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.		
Template Description		
InvalidationPolicy: Specifies whether the component can actively invalidate a particular dataElement. If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.		
ISignalPort.handleInvalid: This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.		
M2 Parameter		
SWComponentTemplate::PortInterface::InvalidationPolicy, SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.handleInvalid		
Mapping Rule		Mapping Type
If strategy HandleInvalidEnum.keep is defined then set ComDataInvalidAction to NOTIFY. If strategy HandleInvalidEnum.replace is defined then set ComDataInvalidAction to REPLACE. In all other cases the ComDataInvalidAction shall not be configured.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00314]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		BSW Type

ComErrorNotification	EcucFunctionNameDef
BSW Description	
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.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00499]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComFilter	EcucParamConfContainerDef
BSW Description	
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.	
Template Description	
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.	
M2 Parameter	
CommonStructure::Filter::DataFilter	
Mapping Rule	
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.	
Mapping Type	
full	
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00339]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal/ComFilter
BSW Parameter	BSW Type
ComFilterAlgorithm	EcucEnumerationParamDef
BSW Description	
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.	
Template Description	
This attribute specifies the type of the filter.	
M2 Parameter	
CommonStructure::Filter::DataFilter.dataFilterType	
Mapping Rule	
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.	
Mapping Type	
full	
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00146]

BSW Module	BSW Context
-------------------	--------------------

Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		BSW Type
ComFilterMask		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Mask for old and new value.		
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00235]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		BSW Type
ComFilterMax		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the upper boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00317]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		BSW Type
ComFilterMin		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the lower boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00318]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		BSW Type

ComFilterOffset	EcucIntegerParamDef
BSW Description	
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.	
Range = 0..(ComFilterPeriod-1)	
Template Description	
Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter	
CommonStructure::Filter::DataFilter.offset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00313]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal/ComFilter
BSW Parameter	BSW Type
ComFilterPeriod	EcucIntegerParamDef
BSW Description	
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.	
Template Description	
Specifies number of messages to occur before the message is passed again	
M2 Parameter	
CommonStructure::Filter::DataFilter.period	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00312]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal/ComFilter
BSW Parameter	BSW Type
ComFilterX	EcucIntegerParamDef
BSW Description	
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.	
Template Description	
Value to compare with	
M2 Parameter	
CommonStructure::Filter::DataFilter.x	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00147]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComFirstTimeout	EcucFloatParamDef

BSW 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.	
Template Description	
-ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. -ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.firstTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00183]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComHandleId	EcucIntegerParamDef
BSW 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.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00165]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComInitialValueOnly	EcucBooleanParamDef
BSW 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 RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).	
Template Description	
Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or DataFilters for ISignals need to be specified several ISignalPorts may be created.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort	
Mapping Rule	Mapping Type

Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00811]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComInvalidNotification	EcucFunctionNameDef	
BSW Description		
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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00315]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComNotification	EcucFunctionNameDef	
BSW Description		
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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00498]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComRxDataTimeoutAction	EcucEnumerationParamDef	
BSW Description		
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
Template Description		
This attribute controls the behavior with respect to the handling of timeouts.		

M2 Parameter	
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.handleTimeoutType	
Mapping Rule	Mapping Type
<p>If a full DataMapping exists for the SystemSignal and there is a single receiver on this ECU then this information shall be configured in accordance with the configured NonqueuedReceiverComSpec.</p> <p>If a full DataMapping exists for the SystemSignal and there are multiple receivers on this ECU then this information is available in the NonqueuedReceiverComSpecs. In this case the attribute ComRxDataTimeoutAction of the related ComSignal/ComSignalGroup shall be configured to NONE to ensure that the RTE always has access to the last received value. Please note that the SWS_RTE defines an algorithm to implement the applicable timeout action.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00412]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComSignalDataInvalidValue	EcucStringParamDef
BSW Description	
<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.</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>	
Template Description	
<p>InvalidationPolicy: Specifies whether the component can actively invalidate a particular dataElement.</p> <p>If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.</p> <p>SwDataDefProps.invalidValue: Optional value to express invalidity of the actual data element.</p>	
M2 Parameter	
SWComponentTemplate::PortInterface::InvalidationPolicy, DataDictionary::DataDefProperties::SwDataDefProps.invalidValue	
Mapping Rule	Mapping Type
<p>ComSignalDataInvalidValue is only derived 1:1 from the SwDataDefProps.invalidValue if the InvalidationPolicy equals keep or replace.</p> <p>In all other cases of InvalidationPolicy the ComSignalDataInvalidValue shall not be configured.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00391]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		BSW Type
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
Template Description		
<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>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByteOrder		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00157]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		BSW Type
ComSignalInitValue		EcucStringParamDef
BSW Description		
<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.</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>		
Template Description		
Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue		
Mapping Rule		Mapping Type
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.		full
Mapping Status		ECUC Parameter ID

valid	[ECUC_Com_00170]
-------	------------------

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComSignalLength	EcucIntegerParamDef	
BSW Description		
<p>Description:</p> <p>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>		
Template Description		
<p>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.</p> <p>The ISignal length of zero bits is allowed.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length		
Mapping Rule		Mapping Type
ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00437]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComSignalType	EcucEnumerationParamDef	
BSW Description		
<p>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.</p>		
Template Description		
<p>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.</p> <p>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.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.dataTypePolicy		
Mapping Rule		Mapping Type

<p>The mapping depends from the setting in the ISignal.dataTypePolicy:</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationDataType.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_02079 and access SwDataDefProps.swBaseType. <p>Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => ComSignalType should not be configured.</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if SystemSignal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true 		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00127]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter	BSW Type	
ComSystemTemplateSystemSignalRef	EcucForeignReferenceDef	
BSW Description		
Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Com_00002]
-------	------------------

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComTimeout	EcucFloatParamDef
BSW Description	
<p>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.</p>	
Template Description	
<p>ISignalPort.timeout: - ISignalPort with communicationDirection = in:</p> <p>Optional timeout value in seconds for the reception of the ISignal in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec, in this case the timeout value in ReceiverComSpec overrides this optional timeout specification.</p> <p>- ISignalPort with communicationDirection = out:</p> <p>Optional timeout value in seconds for transmission deadline monitoring in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec, in this case the timeout value in SenderComSpec overrides this optional timeout specification.</p> <p>TransmissionAcknowledgementRequest.timeout: Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.</p> <p>NonqueuedReceiverComSpec.aliveTimeout: Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description.</p> <p>If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.</p>	
M2 Parameter	
<p>SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.timeout, SW SWComponentTemplate::Communication::TransmissionAcknowledgementRequest.timeout, SW ComponentTemplate::Communication::NonqueuedReceiverComSpec.aliveTimeout</p>	
Mapping Rule	Mapping Type

<p>TX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec.transmissionAcknowledge.timeout that specifies the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been transmitted according to the specified timing description. In this case the timeout value in SenderComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort.</p> <p>RX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec.aliveTimeout. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. Please note that the SWS_RTE defines an algorithm to finally set the applicable timeout value.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00263]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComTimeoutNotification	EcucFunctionNameDef
BSW Description	
On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00552]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComTimeoutSubstitutionValue	EcucStringParamDef
BSW Description	

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.

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.

Template Description

Defines and enables the ComTimeoutSubstitution for this ISignal.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.timeoutSubstitutionValue

Mapping Rule

The mapping of ComTimeoutSubstitutionValue depends on the setting in the ISignal.dataTypePolicy:

- ISignal.dataTypePolicy = override or legacy:
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.timeoutSubstitutionValue
- ISignal.dataTypePolicy = networkRepresentationFromComSpec:
SWComponentTemplate::Communication::NonequeuedReceiverComSpec.timeoutSubstitutionValue
- ISignal.dataTypePolicy = transformingISignal
this is not supported.

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_Com_10006]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComTransferProperty	EcuEnumerationParamDef
BSW 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.	
Template Description	
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00232]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	BSW Type
ComUpdateBitPosition	EcucIntegerParamDef
BSW Description	
<p>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.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>	
Template Description	
<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 SignalPdu. 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 ISignalPdu 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>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndicationBitPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00257]

BSW Module	BSW Context
Com	Com/ComConfig
BSW Parameter	BSW Type
ComSignalGroup	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's signal groups.	
Template Description	

SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignallPdus to multiple receivers.	
An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignal-Group represents a COM Signal Group.	
Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalGroup	
Mapping Rule	Mapping Type
A ComSignalGroup container shall be created for an ISignalGroup that is contained in an ISignallPdu which the Com module is sending.	full
The creation of a ComSignalGroup container may be omitted for an ISignalGroup that is contained in an ISignallPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignalGroup.	
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00345]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	BSW Type
ComDataInvalidAction	EcucEnumerationParamDef
BSW 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.	
Template Description	
InvalidationPolicy: Specifies whether the component can actively invalidate a particular dataElement.	
If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.	
ISignalPort.handleInvalid: This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.	
M2 Parameter	
SWComponentTemplate::PortInterface::InvalidationPolicy, SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.handleInvalid	
Mapping Rule	Mapping Type
If strategy HandleInvalidEnum.keep is defined then set ComDataInvalidAction to NOTIFY. If strategy HandleInvalidEnum.replace is defined then set ComDataInvalidAction to REPLACE. In all other cases the ComDataInvalidAction shall not be configured.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00314]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	BSW Type

ComErrorNotification	EcucFunctionNameDef
BSW Description	
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.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00499]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	BSW Type
ComFirstTimeout	EcucFloatParamDef
BSW 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.	
Template Description	
-ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. -ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.firstTimeout	
Mapping Rule	
Mapping Type	
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00183]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	BSW Type
ComGroupSignal	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.	
Template Description	

Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignallPdus to multiple receivers.	
To support the RTE "signal fan-out" each SignallPdu contains ISignals. If the same System Signal is to be mapped into several SignallPdus 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.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal	
Mapping Rule	Mapping Type
A ComGroupSignal container shall be created for an ISignal contained in an ISignalGroup that is contained in an ISignallPdu which the Com module is sending. The creation of a ComGroupSignal container may be omitted for an ISignal contained in an ISignalGroup that is contained in an ISignallPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal of the ISignal Group.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00520]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	BSW Type
ComBitPosition	EcucIntegerParamDef
BSW 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	
Template Description	
This parameter is necessary to describe the bitposition of a signal within an SignallPdu. 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.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00259]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComBitSize	EcucIntegerParamDef	
BSW 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.		
Template Description		
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.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00158]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComFilter	EcucParamConfContainerDef	
BSW Description		
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.		
Template Description		
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.		
M2 Parameter		
CommonStructure::Filter::DataFilter		
Mapping Rule	Mapping Type	
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00339]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterAlgorithm	EcucEnumerationParamDef	
BSW Description		
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.		
Template Description		
This attribute specifies the type of the filter.		
M2 Parameter		
CommonStructure::Filter::DataFilter.dataFilterType		
Mapping Rule	Mapping Type	

Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00146]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterMask	EcucIntegerParamDef	
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Mask for old and new value.		
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00235]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterMax	EcucIntegerParamDef	
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the upper boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00317]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterMin	EcucIntegerParamDef	
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to specify the lower boundary		
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00318]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterOffset	EcucIntegerParamDef	
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Range = 0..(ComFilterPeriod-1)		
Template Description		
Specifies the initial number of messages to occur before the first message is passed		
M2 Parameter		
CommonStructure::Filter::DataFilter.offset		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00313]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterPeriod	EcucIntegerParamDef	
BSW Description		
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.		
Template Description		
Specifies number of messages to occur before the message is passed again		
M2 Parameter		
CommonStructure::Filter::DataFilter.period		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00312]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter	BSW Type	
ComFilterX	EcucIntegerParamDef	
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
Template Description		
Value to compare with		
M2 Parameter		
CommonStructure::Filter::DataFilter.x		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Com_00147]
-------	------------------

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComHandleId	EcucIntegerParamDef	
BSW 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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00165]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComSignalDataInvalidValue	EcucStringParamDef	
BSW Description		
Defines the data invalid value of the signal.		
<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>		
Template Description		
<p>InvalidationPolicy: Specifies whether the component can actively invalidate a particular dataElement.</p> <p>If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.</p> <p>SwDataDefProps.invalidValue: Optional value to express invalidity of the actual data element.</p>		
M2 Parameter		
SWComponentTemplate::PortInterface::InvalidationPolicy, DataDictionary::DataDefProperties::SwDataDefProps.invalidValue		
Mapping Rule		Mapping Type

ComSignalDataInvalidValue is only derived 1:1 from the SwDataDef Props.invalidValue if the InvalidationPolicy equals keep or replace. In all other cases of InvalidationPolicy the ComSignalDataInvalidValue shall not be configured.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00391]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComSignalEndianness	EcucEnumerationParamDef	
BSW Description		
Defines the endianness of the signal's network representation.		
Template Description		
This parameter defines the order of the bytes of the signal and the packing into the SignalPdu. 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 SignalPdu (see the startPosition attribute description).		
For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByteOrder		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00157]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComSignalInitValue	EcucStringParamDef	
BSW 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.		
Template Description		
Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.		
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue	
Mapping Rule	Mapping Type
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00170]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	BSW Type
ComSignalLength	EcucIntegerParamDef
BSW Description	
<p>Description:</p> <p>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>	
Template Description	
<p>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.</p> <p>Indicates maximum size for dynamic length signals.</p> <p>The ISignal length of zero bits is allowed.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length	
Mapping Rule	Mapping Type
ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00437]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	BSW Type
ComSignalType	EcucEnumerationParamDef
BSW Description	
<p>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.</p> <p>This type could also be used to reserved appropriate storage in AUTOSAR COM.</p>	
Template Description	

<p>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.</p> <p>If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used.</p> <p>If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps.</p> <p>In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.dataTypePolicy	
Mapping Rule	Mapping Type
<p>The mapping depends from the setting in the ISignal.dataTypePolicy:</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.compositeNetworkRepresentation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationDataType.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_02079 and access SwDataDefProps.swBaseType. <p>Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => ComSignalType should not be configured.</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if SystemSignal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true 	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00127]

BSW Module	BSW Context
-------------------	--------------------

Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		BSW Type
ComSystemTemplateSystemSignalRef		EcucForeignReferenceDef
BSW Description		
Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignallPdus and defines the position of the ISignal within an ISignallPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00002]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		BSW Type
ComTimeoutSubstitutionValue		EcucStringParamDef
BSW Description		
<p>The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE.</p> <p>In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00.</p> <p>In case of UIN8_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, UIN8_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 UIN8_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>		
Template Description		
Defines and enables the ComTimeoutSubstitution for this ISignal.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.timeoutSubstitutionValue		
Mapping Rule		Mapping Type

<p>The mapping of ComTimeoutSubstitutionValue depends on the setting in the ISignal.dataTypePolicy:</p> <ul style="list-style-type: none"> - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.timeoutSubstitutionValue - ISignal.dataTypePolicy = networkRepresentationFromComSpec: SWComponentTemplate::Communication::NonequeuedReceiverComSpec.timeoutSubstitutionValue - ISignal.dataTypePolicy = transformingISignal this is not supported. 		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10006]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter	BSW Type	
ComTransferProperty	EcucEnumerationParamDef	
BSW 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.		
Template Description		
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00560]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter	BSW Type	
ComHandleId	EcucIntegerParamDef	
BSW 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.		
Template Description		
M2 Parameter		
Mapping Rule		
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00165]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		BSW Type
ComInitialValueOnly		EcucBooleanParamDef
BSW 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 RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).		
Template Description		
Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or DataFilters for ISignals need to be specified several ISignalPorts may be created.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort		
Mapping Rule		Mapping Type
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00811]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		BSW Type
ComInvalidNotification		EcucFunctionNameDef
BSW Description		
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.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00315]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		BSW Type
ComNotification		EcucFunctionNameDef
BSW Description		
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.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00498]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter	BSW Type	
ComRxDataTimeoutAction	EcucEnumerationParamDef	
BSW Description		
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
Template Description		
This attribute controls the behavior with respect to the handling of timeouts.		
M2 Parameter		
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.handleTimeoutType		
Mapping Rule	Mapping Type	
If a full DataMapping exists for the SystemSignal and there is a single receiver on this ECU then this information shall be configured in accordance with the configured NonqueuedReceiverComSpec.	full	
If a full DataMapping exists for the SystemSignal and there are multiple receivers on this ECU then this information is available in the NonqueuedReceiverComSpecs. In this case the attribute ComRxDataTimeoutAction of the related ComSignal/ComSignalGroup shall be configured to NONE to ensure that the RTE always has access to the last received value. Please note that the SWS_RTE defines an algorithm to implement the applicable timeout action.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00412]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter	BSW Type	
ComSignalGroupArrayAccess	EcucBooleanParamDef	
BSW Description		
Defines whether the uint8-array based access shall be used for this ComSignalGroup.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalGroup.comBasedSignalGroupTransformation		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10003]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter	BSW Type	
ComSystemTemplateSignalGroupRef	EcucForeignReferenceDef	
BSW Description		

Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00001]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	BSW Type
ComTimeout	EcucFloatParamDef
BSW Description	
<p>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.</p>	
Template Description	
<p>ISignalPort.timeout:</p> <p>- ISignalPort with communicationDirection = in:</p> <p>Optional timeout value in seconds for the reception of the ISignal in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec, in this case the timeout value in ReceiverComSpec overrides this optional timeout specification.</p> <p>- ISignalPort with communicationDirection = out:</p> <p>Optional timeout value in seconds for transmission deadline monitoring in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec, in this case the timeout value in SenderComSpec overrides this optional timeout specification.</p> <p>TransmissionAcknowledgementRequest.timeout: Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.</p> <p>NonqueuedReceiverComSpec.aliveTimeout: Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description.</p> <p>If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.timeout, SW SWComponentTemplate::Communication::TransmissionAcknowledgementRequest.timeout, SW ComponentTemplate::Communication::NonqueuedReceiverComSpec.aliveTimeout	

Mapping Rule		Mapping Type
<p>TX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec.transmissionAcknowledge.timeout that specifies the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been transmitted according to the specified timing description. In this case the timeout value in SenderComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort.</p> <p>RX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec.aliveTimeout. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. Please note that the SWS_RTE defines an algorithm to finally set the applicable timeout value.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00263]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		BSW Type
ComTimeoutNotification		EcucFunctionNameDef
BSW Description		
On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00552]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		BSW Type
ComTransferProperty		EcucEnumerationParamDef
BSW 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.		
Template Description		
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00232]	

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter	BSW Type	
ComUpdateBitPosition	EcucIntegerParamDef	
BSW Description		
<p>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.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>		
Template Description		
<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 SignalPdu. 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 ISignalPdu 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>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndicationBitPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00257]

BSW Module	BSW Context	
Com	Com	
BSW Parameter	BSW Type	
ComGeneral	EcucParamConfContainerDef	
BSW Description		
Contains the general configuration parameters of the module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Com_00541]

BSW Module		BSW Context	
Com		Com/ComGeneral	
BSW Parameter		BSW Type	
ComCancellationSupport		EcucBooleanParamDef	
BSW Description			
This parameter enables/disables the cancellation feature: true: enabled false: disabled			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Com_10000]	

BSW Module		BSW Context	
Com		Com/ComGeneral	
BSW Parameter		BSW Type	
ComConfigurationUseDet		EcucBooleanParamDef	
BSW Description			
The error hook shall contain code to call the Det. If this parameter is configured COM_DEV_ERROR_DETECT shall be set to ON as output of the configuration tool. (as input for the source code).			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Com_00141]	

BSW Module		BSW Context	
Com		Com/ComGeneral	
BSW Parameter		BSW Type	
ComEnableMDTForCyclicTransmission		EcucBooleanParamDef	
BSW Description			
Enables globally for the whole Com module the minimum delay time monitoring for cyclic and repeated transmissions (ComTxModeMode=PERIODIC or ComTxModeMode=MIXED for the cyclic transmissions, ComTxModeNumberOfRepetitions > 0 for repeated transmissions).			
Template Description			
Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions > 0).			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.comEnableMDTForCyclicTransmission			
Mapping Rule		Mapping Type	
1:1 mapping		full	

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00788]

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter	BSW Type	
ComEnableSignalGroupArrayApi	EcucBooleanParamDef	
BSW Description		
Activate/Deactivate the signal group array access APIs (Com_SendSignalGroupArray, Com_ReceiveSignalGroupArray).		
true: signal group array access APIs activated		
false: signal group array access APIs deactivated		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10002]	

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter	BSW Type	
ComMetaDataSupport	EcucBooleanParamDef	
BSW Description		
This parameter enables/disables the support of meta-data feature including the API Com_TriggerIPDUSendWithMetaData.		
true: enabled		
false: disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10004]	

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter	BSW Type	
ComRetryFailedTransmitRequests	EcucBooleanParamDef	
BSW Description		
If this Parameter is set to true, retry of failed transmission requests is enabled. If this Parameter is not present, the default value is assumed.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00780]	

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		BSW Type
ComSupportedIPduGroups		EcucIntegerParamDef
BSW Description		
Defines the maximum number of supported I-PDU groups.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_00710]	

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		BSW Type
ComUserCbKHeaderFile		EcucStringParamDef
BSW Description		
Defines the header files for callback functions which shall be included by the COM module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Com_10001]	

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		BSW Type
ComVersionInfoApi		EcucBooleanParamDef
BSW Description		
Activate/Deactivate the version information API (Com_GetVersionInfo).		
True: version information API activated False: version information API deactivated		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00438]

C.1.2 LdCom Mapping

BSW Module	BSW Context	
LdCom	LdCom	
BSW Parameter	BSW Type	
LdComConfig	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR LdCom module.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LdCom_00003]	

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig	
BSW Parameter	BSW Type	
LdComIPdu	EcucParamConfContainerDef	
BSW Description		
Contains the configuration parameters of the IPdu inside LdCom.		
Template Description		
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.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LdCom_00006]	

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter	BSW Type	
LdComApiType	EcucEnumerationParamDef	
BSW Description		
Defines if this I-PDU is a normal I-PDU that shall be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.		
This setting is used by RTE to invoke the proper API.		

Template Description	
Contains all configuration elements for AUTOSAR TP.	
M2 Parameter	
SystemTemplate::TransportProtocols::TpConfig	
Mapping Rule	Mapping Type
If this LdComIPdu is mapped in the System Description by a TpConnection to NPdus then set LdComApiType to TP. Otherwise set LdComApiType to IF.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00002]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComHandleId	EcucIntegerParamDef
BSW Description	
This is the ID used by RTE to invoke LdCom. A corresponding shortName is created, which is used for the invocations of the RTE. The same ID is used for invocations by PduR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00005]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComIPduDirection	EcucEnumerationParamDef
BSW Description	
The direction defines if this IPdu, and therefore the contributing signal, shall be sent or received.	
Template Description	
Communication Direction of the Connector Port (input or output Port).	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommConnectorPort.communicationDirection	
Mapping Rule	Mapping Type
Find IPduTriggering of the regarded SignallIPdu. The IPduTriggering contains a reference to an IPduPort that is aggregated by the regarded ECU. If the communicationDirection of the CommConnectorPort is "in" than the IPdu is received.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00007]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComPduRef	EcucReferenceDef
BSW Description	
Reference to the global Pdu.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00010]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComRxCopyRxData	EcucFunctionNameDef
BSW Description	
Only on receiver side: Name of Rte_LdComCbkJCopyRxData callback function to be called.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00013]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComRxIndication	EcucFunctionNameDef
BSW Description	
Only on receiver side: Name of Rte_LdComCbkJRxIndication callback function to be called.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00014]

BSW Module	BSW Context
LdCom	LdCom/LdComConfig/LdComIPdu
BSW Parameter	BSW Type
LdComRxStartOfReception	EcucFunctionNameDef
BSW Description	
Only on receiver side: Name of Rte_LdComCbkJStartOfReception callback function to be called.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00015]

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter	BSW Type	
LdComSystemTemplateSignalRef	EcucForeignReferenceDef	
BSW Description		
Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template).		
Template Description		
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LdCom_00011]	

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter	BSW Type	
LdComTpRxIndication	EcucFunctionNameDef	
BSW Description		
Only on receiver side: Name of Rte_LdComCbKTpRxIndication callback function to be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LdCom_00016]	

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter	BSW Type	
LdComTpTxConfirmation	EcucFunctionNameDef	
BSW Description		
Only on sender side: Name of Rte_LdComCbKTpTxConfirmation callback function to be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LdCom_00017]	

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter		BSW Type
LdComTxConfirmation		EcucFunctionNameDef
BSW Description		
Only on sender side: Name of Rte_LdComCbkJTxConfirmation callback function to be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LdCom_00021]

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter		BSW Type
LdComTxCopyTxData		EcucFunctionNameDef
BSW Description		
Only on sender side: Name of Rte_LdComCbkJCopyTxData callback function to be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LdCom_00018]

BSW Module	BSW Context	
LdCom	LdCom/LdComConfig/LdComIPdu	
BSW Parameter		BSW Type
LdComTxTriggerTransmit		EcucFunctionNameDef
BSW Description		
Only on sender side: Name of Rte_LdComCbkJTriggerTransmit callback function to be called. If defined TriggerTransmit has to be supported for this signal.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LdCom_00019]

BSW Module	BSW Context	
LdCom	LdCom	
BSW Parameter		BSW Type
LdComGeneral		EcucParamConfContainerDef

BSW Description	
Contains the general configuration parameters of the LdCom module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00004]

BSW Module	BSW Context
LdCom	LdCom/LdComGeneral
BSW Parameter	BSW Type
LdComDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00020]

BSW Module	BSW Context
LdCom	LdCom/LdComGeneral
BSW Parameter	BSW Type
LdComVersionInfoApi	EcucBooleanParamDef
BSW Description	
Activate/Deactivate the version information API (LdCom_GetVersionInfo). * True: version information API activated * False: version information API deactivated	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LdCom_00012]

C.1.3 IPduM Mapping

BSW Module	BSW Context
IpduM	IpduM
BSW Parameter	BSW Type

IpduMConfig	EcucParamConfContainerDef
BSW Description	
<p>This container contains the sub containers of the IpduM module.</p> <ul style="list-style-type: none"> * The IpduMTxPathway subcontainer includes information about sent I-PDUs. * The IpduMRxPathway includes information about received I-PDUs. * The IpduMContainerTxPdu and IpduMContainedTxPdu include information about the sending of ContainerPdu. * The IpduMContainerRxPdu and IpduMContainedRxPdu include information about the reception of ContainerPdu. 	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_IpduM_00059]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMContainedRxPdu	EcucParamConfContainerDef
BSW Description	
Configuration of a received contained Pdu.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_IpduM_00174]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	BSW Type
IpduMContainedPduOffset	EcucIntegerParamDef
BSW Description	
Static offset (in bytes) of the ContainedPdu.	
Template Description	
Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.offset	
Mapping Rule	
1:1 mapping	Mapping Type
	full
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_IpduM_00206]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu

BSW Parameter		BSW Type
IpduMContainedRxInContainerPduRef		EcucReferenceDef
BSW Description		
Optional reference to an IpduMContainerRxPdu this IpduMContainedRxPdu may be received in.		
If this IpduMContainedRxPdu shall be received in exactly one IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_CONFIGURED then the IpduMContainedRxInContainerPduRef shall be defined.		
If this IpduMContainedRxPdu can be received in any IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL then the IpduMContainedRxInContainerPduRef shall NOT be defined.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.containedPduTriggering		
Mapping Rule		Mapping Type
In the SysT the ContainerPdu references all PduTriggerings which can be put inside this container. In the EcuC each Pdu refers to the containers it can be transported in.		partial
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00173]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu	
BSW Parameter		BSW Type
IpduMContainedRxPduLongHeaderId		EcucIntegerParamDef
BSW Description		
LongHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Template Description		
Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdLongHeader		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00203]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu	
BSW Parameter		BSW Type
IpduMContainedRxPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu which represents this ContainedPdu and is used for reception indication.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00175]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu	
BSW Parameter	BSW Type	
IpduMContainedRxPduShortHeaderId	EcucIntegerParamDef	
BSW Description		
ShortHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Template Description		
Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdShortHeader		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00202]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu	
BSW Parameter	BSW Type	
IpduMPduUpdateBitPosition	EcucIntegerParamDef	
BSW Description		
This value specifies where the PDU's Update-Bit is stored in the Container PDU (bit location of PDU's Update-Bit in the Container PDU).		
Template Description		
The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.updateIndicationBitPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00207]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter	BSW Type	
IpduMContainedTxPdu	EcucParamConfContainerDef	
BSW Description		
Configuration of a sender ContainedPdu.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00177]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedPduHeaderId	EcucIntegerParamDef
BSW Description	
Header Id which is part of the ContainerPdu when this ContainedPdu is inside.	
Template Description	
ContainedIPduProps.headerIdLongHeader: Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.	
ContainedIPduProps.headerIdShortHeader: Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdLongHeader, SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdShortHeader	
Mapping Rule	Mapping Type
If IpduMContainerHeaderSize = LONG the IPduMContainedPduHeaderId is taken from headerIdLongHeader. If IpduMContainerHeaderSize = SHORT the IPduMContainedPduHeaderId is taken from headerIdShortHeader.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00172]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedPduOffset	EcucIntegerParamDef
BSW Description	
Static offset (in bytes) of the ContainedPdu.	
Template Description	
Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.offset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00206]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxInContainerPduRef	EcucReferenceDef
BSW Description	
Reference to the container Pdu which this contained Pdu shall be collected in.	
Template Description	

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.containedPduTriggering	
Mapping Rule	Mapping Type
In the SysT the ContainerPdu references all PduTriggerings which can be put inside this container. In the EcuC each Pdu refers to the containers it can be transported in.	partial
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00176]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxPduCollectionSemantics	EcucEnumerationParamDef
BSW Description	
Defines whether this IpduMContainedTxPdu shall be collected using a last-is-best or queued semantics.	
Template Description	
Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.collectionSemantics	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00198]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduCollectionSemantics
BSW Parameter	BSW Type
IPDUM_COLLECT_LAST_IS_BEST	EcucEnumerationLiteralDef
BSW Description	
The IpduMContainedTxPdu data will be fetched via TriggerTransmit just before the transmission executes.	
Template Description	
The ContainedIPdu data will be fetched via TriggerTransmit just before the transmission executes.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduCollectionSemanticsEnum.lastIsBest	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduCollectionSemantics
BSW Parameter	BSW Type
IPDUM_COLLECT_QUEUED	EcucEnumerationLiteralDef
BSW Description	

The IpduMContainedTxPdu data will instantly be stored to the IpduMContainerTxPdu in the context of the Transmit API.	
Template Description	
The ContainedIPdu data will instantly be stored to the ContainerIPdu in the context of the Transmit API.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduCollectionSemantics Enum.queued	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxPduConfirmation	EcucBooleanParamDef
BSW Description	
This Parameter determines whether for this contained I-PDU a TxConfirmation shall be provided. If set to TRUE a TxConfirmation is issued. It is not used when an I-PDU is requested using the trigger transmit API.	
If this Parameter is omitted, the default value shall be used.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00178]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxPduHandleId	EcucIntegerParamDef
BSW Description	
Handle Id of the ContainedPdu.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00179]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type

IpduMContainedTxPduPriority	EcucIntegerParamDef
BSW Description	
Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.	
Template Description	
Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.priority	
Mapping Rule	Mapping Type
Priority handling for a ContainerIPdu is enabled if at least one ContainedIPdu contains the attribute "priority" within its aggregated ContainerIPduProps, and there are different priorities configured within one ContainerIPdu (Reason: When all ContainedIPdus have the same priority, they cannot be prioritized).	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00210]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu which represents this ContainedPdu and is used for transmission.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00180]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	BSW Type
IpduMContainedTxPduSendTimeout	EcucFloatParamDef
BSW Description	
Defines a ContainedPdu specific sender timeout which can reduce the ContainerPdu timer when this ContainedPdu is put inside the ContainerPdu. Defined in seconds.	
Template Description	
Defines a IPdu specific sender timeout which can reduce the ContainerIPdu timer when this containedIPdu is put inside the ContainerIPdu. This attribute is ignored on receiver side.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.timeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00181]

BSW Module		BSW Context	
IpduM		IpduM/IpduMConfig/IpduMContainedTxPdu	
BSW Parameter		BSW Type	
IpduMContainedTxPduTrigger		EcucEnumerationParamDef	
BSW Description			
Defines whether this Pdu triggers the sending of the ContainerPdu.			
Template Description			
Defines whether this IPdu does trigger the sending of the ContainerIPdu. This attribute is ignored on receiver side.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.trigger			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_IpduM_00182]	

BSW Module		BSW Context	
IpduM		IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduTrigger	
BSW Parameter		BSW Type	
IPDUM_TRIGGER_ALWAYS		EcucEnumerationLiteralDef	
BSW Description			
This Pdu directly triggers the sending of the ContainerPdu.			
Template Description			
Pdu will trigger the transmission of the data.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduCollectionTriggerEnum.always			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
IpduM		IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduTrigger	
BSW Parameter		BSW Type	
IPDUM_TRIGGER_NEVER		EcucEnumerationLiteralDef	
BSW Description			
This Pdu does not triggers the sending of the ContainerPdu (other trigger criteria might still trigger sending of the ContainerPdu).			
Template Description			
Pdu will be buffered and will not trigger the transmission of the data.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduCollectionTriggerEnum.never			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
IpduM		IpduM/IpduMConfig/IpduMContainedTxPdu	
BSW Parameter		BSW Type	

IpduMPduUpdateBitPosition	EcucIntegerParamDef
BSW Description	
This value specifies where the PDU's Update-Bit is stored in the Container PDU (bit location of PDU's Update-Bit in the Container PDU).	
Template Description	
The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.updateIndicationBit Position	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00207]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMContainerRxPdu	EcucParamConfContainerDef
BSW Description	
Configuration of a receiver ContainerPdu which may collect several ContainedPdus.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00188]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	BSW Type
IpduMContainerHeaderSize	EcucEnumerationParamDef
BSW Description	
Defines the layout of the header information (header id and length).	
Template Description	
Defines whether and which header type is used (header id and length).	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.headerType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00183]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_LONG	EcucEnumerationLiteralDef
BSW Description	

Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
Template Description	
Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.long Header	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_NONE	EcucEnumerationLiteralDef
BSW Description	
Static Container Layout	
Template Description	
No Header is used and the location of each containedPdu in the ContainerPdu is statically configured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.no Header	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_SHORT	EcucEnumerationLiteralDef
BSW Description	
Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit	
Template Description	
Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.short Header	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu	
BSW Parameter	BSW Type	
IpduMContainerPduProcessing	EcucEnumerationParamDef	
BSW Description		
Defines whether the handling of this ContainerPdu shall be done in the context of the caller (IMMEDIATE) or in the next call to IpduM_MainFunctionRx (DEFERRED).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00184]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu	
BSW Parameter	BSW Type	
IpduMContainerQueueSize	EcucIntegerParamDef	
BSW Description		
Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.		
Template Description		
ContainerIPdu.minimumRxContainerQueueSize: This attribute defines the minimum queue size for received containers.		
ContainerIPdu.minimumTxContainerQueueSize: This attribute defines the minimum queue size for transmitted containers.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.minimumRxContainerQueueSize, SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.minimumTxContainerQueueSize		
Mapping Rule		Mapping Type
The value of this parameter can not be derived from the System Extract but the System Extract may define a minimum queue size. If this parameter is used in the context of the IpduMContainerTxPdu then the minimumTxContainerQueueSize attribute needs to be respected. If this parameter is used in the context of the IpduMContainerRxPdu then the minimumRxContainerQueueSize attribute needs to be respected.		partial
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00185]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu	
BSW Parameter	BSW Type	
IpduMContainerRxAcceptContainedPdu	EcucEnumerationParamDef	
BSW Description		
Defines for the received IpduMContainerRxPdu whether the list of referencing IpduMContainedRxPdus (via the reference IpduMContainedPduContainerRefRx) is a closed set.		
Template Description		
Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.		

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.rxAcceptContainedIPdu	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00186]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerRxAcceptCon- tainedPdu
BSW Parameter	BSW Type
IPDUM_ACCEPT_ALL	EcucEnumerationLiteralDef
BSW Description	
The IpduMContainedRxPdus which are referencing this IpduMContainerRxPdu are expected inside this IpduMContainerRxPdu, but there may also occur other Pdus inside this IpduMContainerRxPdu as well. This also supports the case where no IpduMContainedRxPdu references the IpduMContainerRxPdu.	
Template Description	
No fixed set of containedIPdus is defined for reception, any known containedIPdu (based on headerId) shall be expected within this ContainerIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::RxAcceptContainedIPduEnum.acceptAll	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerRxAcceptCon- tainedPdu
BSW Parameter	BSW Type
IPDUM_ACCEPT_CONFIGURED	EcucEnumerationLiteralDef
BSW Description	
Only the IpduMContainedRxPdus which are referencing this IpduMContainerRxPdu are expected inside this IpduMContainerRxPdu.	
Template Description	
A fixed set of containedIPdus is defined for reception. Only these assigned containedIPdus (based on headerId) are expected in this ContainerIPdu. If a not assigned containedIPdu is received within this ContainerIPdu this containedIPdu is discarded.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::RxAcceptContainedIPduEnum.acceptConfigured	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	BSW Type
IpduMContainerRxHandleId	EcucIntegerParamDef

BSW Description	
Handle Id used by the PduR for RxIndication.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00187]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	BSW Type
IpduMContainerRxPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu which represents the container and is used for reception.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00189]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	BSW Type
IpduMMainFunctionRxRef	EcucReferenceDef
BSW Description	
Reference to the IpduM_MainFunctionRx instance this container PDU belongs to.	
Mandatory, if more than one IpduM_MainFunctionRx is defined.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00212]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMContainerTxPdu	EcucParamConfContainerDef
BSW Description	
Configuration of a transmitted container Pdu.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00192]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	BSW Type
IpduMContainerHeaderSize	EcucEnumerationParamDef
BSW Description	
Defines the layout of the header information (header id and length).	
Template Description	
Defines whether and which header type is used (header id and length).	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.headerType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00183]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_LONG	EcucEnumerationLiteralDef
BSW Description	
Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
Template Description	
Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.long Header	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_NONE	EcucEnumerationLiteralDef
BSW Description	
Static Container Layout	
Template Description	

No Header is used and the location of each containedPdu in the ContainerPdu is statically configured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.noHeader	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize
BSW Parameter	BSW Type
IPDUM_HEADERTYPE_SHORT	EcucEnumerationLiteralDef
BSW Description	
Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit	
Template Description	
Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPduHeaderTypeEnum.shortHeader	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	BSW Type
IpduMContainerQueueSize	EcucIntegerParamDef
BSW Description	
Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.	
Template Description	
ContainerIPdu.minimumRxContainerQueueSize: This attribute defines the minimum queue size for received containers.	
ContainerIPdu.minimumTxContainerQueueSize: This attribute defines the minimum queue size for transmitted containers.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.minimumRxContainerQueueSize, SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.minimumTxContainerQueueSize	
Mapping Rule	Mapping Type

<p>The value of this parameter can not be derived from the System Extract but the System Extract may define a minimum queue size. If this parameter is used in the context of the IpduMContainerTxPdu then the minimumTxContainerQueueSize attribute needs to be respected. If this parameter is used in the context of the IpduMContainerRxPdu then the minimumRxContainerQueueSize attribute needs to be respected.</p>		partial
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00185]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter	BSW Type	
IpduMContainerTxFirstContainedPduTrigger	EcucBooleanParamDef	
BSW Description		
Defines if the transmission of this IpduMContainerTxPdu shall be requested right after the first IpduMContainedTxPdu was put into it.		
Template Description		
Defines if the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into it. This attribute shall be ignored on receiver side.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.containerTrigger		
Mapping Rule		Mapping Type
TRUE if ContainerIPdu.containerTrigger = firstContainedTrigger, else FALSE.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00199]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter	BSW Type	
IpduMContainerTxHandleId	EcucIntegerParamDef	
BSW Description		
Handle Id used by the PduR for TxConfirmation and for TriggerTransmit of the ContainerPdu.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00191]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter	BSW Type	
IpduMContainerTxPduRef	EcucReferenceDef	
BSW Description		
Reference to the Pdu which represents the container and is used for transmission.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00193]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter		BSW Type
IpduMContainerTxSendTimeout		EcucFloatParamDef
BSW Description		
When this timeout expires the ContainerPdu is triggered for sending. The respective timer is started when the first Pdu is put into the ContainerPdu. Defined in seconds.		
Template Description		
When this timeout expires the ContainerIPdu is sent out. The respective timer is started when the first Ipdu is put into the ContainerIPdu. This attribute is ignored on receiver side.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.containerTimeout		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00194]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter		BSW Type
IpduMContainerTxSizeThreshold		EcucIntegerParamDef
BSW Description		
Defines the size threshold in bytes which, when exceeded, triggers the sending of the ContainerPdu although the maxium Pdu size (PduLength parameter of Pdu object) has not been reached yet.		
Template Description		
Defines the size threshold which, when exceeded, triggers the sending of the ContainerIPdu although the maxium Pdu size has not been reached yet. Unit: byte.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.thresholdSize		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00195]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu	
BSW Parameter		BSW Type
IpduMContainerTxTriggerMode		EcucEnumerationParamDef
BSW Description		
Defines whether this ContainerPdu is fetched via trigger transmit.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00196]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	BSW Type
IpduMMainFunctionTxRef	EcucReferenceDef
BSW Description	
Reference to the IpduM_MainFunctionTx instance this container PDU belongs to.	
Mandatory, if more than one IpduM_MainFunctionTx is defined.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00214]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	BSW Type
IpduMUnusedAreasDefault	EcucIntegerParamDef
BSW Description	
IpduM fills not updated areas of the Container PDU with this byte-pattern.	
Template Description	
IPduM fills not updated areas of the ContainerPdu with this byte-pattern.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainerIPdu.unusedBitPattern	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00208]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMMainFunctionRx	EcucParamConfContainerDef
BSW Description	
Each element of this container defines one instance IpduM_MainFunctionRx, in case multi-core distribution feature is active.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00211]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMMainFunctionRx	
BSW Parameter		BSW Type
IpduMMainRxPartitionRef		EcucReferenceDef
BSW Description		
Reference to EcucPartition, where the according IpduM_MainFunction instance is assigned to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00215]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMMainFunctionRx	
BSW Parameter		BSW Type
IpduMMainRxTimeBase		EcucFloatParamDef
BSW Description		
<p>The period between successive calls to according instance of IpduM_MainFunctionRx in seconds. This parameter may be used by the IpduM generator to transform the values of the reception related timing configuration parameters of the IpduM module to internal implementation specific counter or tick values. The IpduM module's internal timing handling is implementation specific.</p> <p>The IpduM module (generator) may rely on the fact that IpduM_MainFunctionRx is scheduled according to the value configured here.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00216]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		BSW Type
IpduMMainFunctionTx		EcucParamConfContainerDef
BSW Description		
Each element of this container defines one instance IpduM_MainFunctionTx, in case multi-core distribution feature is active (mutual exclusive to ComTimeBase).		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00213]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMMainFunctionTx
BSW Parameter	BSW Type
IpduMMainTxPartitionRef	EcucReferenceDef
BSW Description	
Reference to EcucPartition, where the according IpduM_MainFunction instance is assigned to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00217]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMMainFunctionTx
BSW Parameter	BSW Type
IpduMMainTxTimeBase	EcucFloatParamDef
BSW Description	
<p>The period between successive calls to IpduM_MainFunctionTx in seconds. This parameter may be used by the IpduM generator to transform the values of the reception related timing configuration parameters of the IpduM module to internal implementation specific counter or tick values. The IpduM module's internal timing handling is implementation specific.</p> <p>The IpduM module (generator) may rely on the fact that IpduM_MainFunctionTx is scheduled according to the value configured here.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00218]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMMaxTxBufferSize	EcucIntegerParamDef
BSW Description	
Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00166]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMMaxTxPathwayCnt	EcucIntegerParamDef
BSW Description	
Maximum number of transmitted IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00165]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	BSW Type
IpduMRxPathway	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters received I-PDUs by the IpduM module.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.	
A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu	
Mapping Rule	Mapping Type
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00071]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway
BSW Parameter	BSW Type

IpduMRxIndication	EcucParamConfContainerDef
BSW Description	
Contains the configuration for incoming RxIndication calls.	
Template Description	
<p>A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.</p> <p>A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu	
Mapping Rule	Mapping Type
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00047]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMByteOrder	EcucEnumerationParamDef
BSW Description	
<p>This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu.</p> <p>The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.</p>	
Template Description	
<p>MultiplexedIPdu.selectorFieldByteOrder: This attribute defines the order of the bytes of the selectorField and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute.</p> <p>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>	
<p>SegmentPosition.segmentByteOrder: This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldByteOrder, SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentByteOrder	
Mapping Rule	Mapping Type

A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00162]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter	BSW Type	
IpduMRxDynamicPart	EcucParamConfContainerDef	
BSW Description		
<p>This container contains the configuration for the dynamic part of incoming RxIndication calls. When an incoming received I-PDU's selector field matches the IpduMRxSelectorValue, the new outgoing I-PDU for the dynamic part is constructed as defined by the segments (defined in the IpduMDynamicSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingDynamicPduRef.</p> <p>In case no dynamic part shall be extracted from this received I-PDU this container does not exist. This use-case can occur in case a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu.</p>		
Template Description		
<p>One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative		
Mapping Rule		Mapping Type
Create container for each DynamicPartAlternative of the MultiplexedIPdu.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00048]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart	
BSW Parameter	BSW Type	
IpduMOutgoingDynamicPduRef	EcucReferenceDef	
BSW Description		
<p>When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent PDU representation in the ECU Configuration Description exchange file.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00112]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart	
BSW Parameter	BSW Type	

IpduMRxSelectorValue	EcucIntegerParamDef
BSW Description	
This is the selector value that this container refers to.	
Template Description	
The selector field is part of a multiplexed IPdu. It consists of contiguous bits. The value of the selector field selects the layout of the multiplexed part of the IPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative.selectorField Code	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00113]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMRxDynamicSegment	EcucParamConfContainerDef
BSW Description	
<p>The dynamic part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments.</p> <p>For each segment one IpduMRxDynamicSegment container shall be created that contains the location and the length of the segment.</p> <p>Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxDynamicPart container and will be copied from the same location in the multiplexed incoming I-Pdu. The segment layout for all dynamic Parts is always identical.</p>	
Template Description	
<p>The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.</p> <p>The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	Mapping Type
Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00170]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamic Segment
BSW Parameter	BSW Type
IpduMSegmentLength	EcucIntegerParamDef
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00114]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicSegment
BSW Parameter	BSW Type
IpduMSegmentPosition	EcucIntegerParamDef
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	
Segments bit position relatively to the beginning of a multiplexed IPdu.	
<p>Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMRxHandleId	EcucIntegerParamDef
BSW Description	
This is the I-PDU ID of the incoming I-PDU. If an incoming RxIndication's I-PDU ID matches this value then it is unpacked according to the specification in this container.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00109]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMRxIndicationPduRef	EcucReferenceDef
BSW Description	
Reference to the received Pdu representation in the ECU Configuration Description exchange file.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00108]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMRxStaticPart	EcucParamConfContainerDef
BSW Description	
This container contains the configuration for the static part of incoming RxIndication calls. On reception, the new outgoing I-PDU for the static part is constructed as defined by the segments (defined in the IpduMStaticSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingStaticPduRef.	
Template Description	
Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::StaticPart	
Mapping Rule	Mapping Type
Create container if StaticPart exists in the MultiplexedIPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00049]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart
BSW Parameter	BSW Type
IpduMOutgoingStaticPduRef	EcucReferenceDef
BSW Description	
When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent Pdu representation in the ECU Configuration Description exchange file.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00115]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMRxStaticSegment	EcucParamConfContainerDef
BSW Description	

<p>The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxStaticSegment container shall be created that contains the location and the length of the segment.</p> <p>Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.</p>	
Template Description	
<p>The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.</p> <p>The ISignalPdu are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalPdu are copied into this first segment and so on.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	Mapping Type
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00169]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment
BSW Parameter	BSW Type
IpduMSegmentLength	EcucIntegerParamDef
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00114]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment
BSW Parameter	BSW Type
IpduMSegmentPosition	EcucIntegerParamDef
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	

Segments bit position relatively to the beginning of a multiplexed IPdu.

Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition

Mapping Rule

1:1 mapping

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	BSW Type
IpduMSelectorField	EcucParamConfContainerDef
BSW Description	
This contains the location and the length of the selector field.	
Template Description	
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
The ISignallPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignallPdu are copied into this first segment and so on.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	Mapping Type
Can be derived from the segmentPosition.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00054]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMSelectorField
BSW Parameter	BSW Type
IpduMSelectorFieldLength	EcucIntegerParamDef
BSW Description	
Length of the selector field in bits.	
Template Description	
The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_IpduM_00160]
-------	--------------------

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMSelectorField	
BSW Parameter		BSW Type
IpduMSelectorFieldPosition		EcucIntegerParamDef
BSW Description		
Selector field bit position in the multiplexed Pdu. Range: 0..63 for CAN/ LIN I-PDUs, 0..511 for CAN FD I-PDUs, 0..2031 for FlexRay I-PDUs.		
Template Description		
This parameter is necessary to describe the position of the selector field within the IPdu. Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorFieldByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldStartPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00161]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		BSW Type
IpduMTxPathway		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters transmitted I-PDUs by the IpduM module.		
Template Description		
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		Mapping Type

Create container for each transmitted multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "Out" Pdu Port.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00070]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway	
BSW Parameter	BSW Type	
IpduMTxRequest	EcucParamConfContainerDef	
BSW Description		
<p>This container is used to specify the configuration for Transmit requests. There will be one instance of this container for each I-PDU that can be requested for transmission (the outgoing I-PDUs) by the IpduM.</p>		
Template Description		
<p>A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.</p> <p>A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		Mapping Type
Create container for each transmitted multiplexed Ipdu		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00052]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter	BSW Type	
IpduMByteOrder	EcucEnumerationParamDef	
BSW Description		
<p>This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu.</p> <p>The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.</p>		
Template Description		

MultiplexedIPdu.selectorFieldByteOrder:

This attribute defines the order of the bytes of the selectorField and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute.

In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.

SegmentPosition.segmentByteOrder:

This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldByteOrder, SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentByteOrder

Mapping Rule

A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_IpduM_00162]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMIPduUnusedAreasDefault	EcucIntegerParamDef
BSW Description	
IpduM module fills not used areas of an I-PDU with this bit-pattern If this attribute is omitted the IpduM module does not fill the I-PDU.	
Template Description	
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.	
In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.unusedBitPattern	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00121]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMInitialDynamicPart	EcucReferenceDef
BSW Description	
Reference to the dynamic part that shall be used to initialize this multiplexed TX-I-PDU.	

Template Description	
Dynamic part that shall be used to initialize this multiplexed IPdu.	
Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative.initialDynamicPart	
Mapping Rule	Mapping Type
If the attribute initialDynamicPart is set to true then create this reference.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00157]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMOutgoingPduRef	EcucReferenceDef
BSW Description	
Reference to the PDU defining the outgoing I-PDU. When the outgoing I-PDU is sent this is the I-PDU ID to give it. It is the IpduM I-PDU ID of the assembled I-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00120]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMSelectorField	EcucParamConfContainerDef
BSW Description	
This contains the location and the length of the selector field.	
Template Description	
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	Mapping Type
Can be derived from the segmentPosition.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00054]

BSW Module	BSW Context
------------	-------------

IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMSelectorField	
BSW Parameter		BSW Type
IpduMSelectorFieldLength	EcucIntegerParamDef	
BSW Description		
Length of the selector field in bits.		
Template Description		
<p>The size in bits of the selector field shall be configurable in a range of 1-16 bits.</p> <p>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00160]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMSelectorField	
BSW Parameter		BSW Type
IpduMSelectorFieldPosition	EcucIntegerParamDef	
BSW Description		
Selector field bit position in the multiplexed Pdu.		
<p>Range: 0..63 for CAN/ LIN I-PDUs, 0..511 for CAN FD I-PDUs, 0..2031 for FlexRay I-PDUs.</p>		
Template Description		
This parameter is necessary to describe the position of the selector field within the IPdu.		
<p>Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorFieldByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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>In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldStartPosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00161]

BSW Module	BSW Context
-------------------	--------------------

IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		BSW Type
IpduMTxConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id used by the PduR for confirmation (IpduM_TxConfirmation) and for TriggerTransmit (IpduM_TriggerTransmit). The existence of this parameter is essential for the PduR generation tool to actually find a symbolic-NameValue for the OutgoingPdu.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00158]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		BSW Type
IpduMTxDynamicPart		EcucParamConfContainerDef
BSW Description		
Configuration parameters for an instance of a TxRequest call into the IpduM. When a Tx Request with the IpduMTxDynamicHandleId is received by the IpduM, all segments (defined in the IpduMDynamicSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used by the dynamic part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the dynamic part.		
Template Description		
One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative		
Mapping Rule		Mapping Type
Create container for each DynamicPartAlternative of the MultiplexedIPdu.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00056]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		BSW Type
IpduMJitUpdate		EcucBooleanParamDef
BSW Description		
If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00167]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter	BSW Type	
IpduMTxDynamicConfirmation	EcucBooleanParamDef	
BSW Description		
A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the dynamic part is generated.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00163]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter	BSW Type	
IpduMTxDynamicHandleId	EcucIntegerParamDef	
BSW Description		
This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured dynamic segments are copied and the IpduMTxTriggerMode is honored.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00127]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter	BSW Type	
IpduMTxDynamicPduRef	EcucReferenceDef	
BSW Description		
Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00126]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		BSW Type
IpduMTxDynamicSegment		EcucParamConfContainerDef
BSW Description		
<p>The dynamic part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxDynamicSegment container shall be created that contains the location and the length of the segment.</p> <p>Please note that each configured segment will be copied out of the source I-Pdu that is referenced in the IpduMTxDynamicPart container and will be copied to the same location in the multiplexed outgoing I-Pdu. The segment layout for all dynamic Parts is always identical.</p>		
Template Description		
<p>The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.</p> <p>The ISignalPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalPdu are copied into this first segment and so on.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		Mapping Type
Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00168]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicSegment	
BSW Parameter		BSW Type
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
Template Description		
Data Length of the segment in bits.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00114]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicSegment	
BSW Parameter		BSW Type
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
Template Description		

Segments bit position relatively to the beginning of a multiplexed IPdu.

Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition

Mapping Rule

1:1 mapping

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMTxStaticPart	EcucParamConfContainerDef
BSW Description	
Configuration parameters for an instance of a Tx_Request call into the IpduM. When a Tx Request with the IpduMTxStaticHandleId is received by the IpduM, all segments (defined in the IpduMStaticSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used for the static part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the static part if it exists.	
Template Description	
Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::StaticPart	
Mapping Rule	Mapping Type
Create container if StaticPart exists in the MultiplexedIPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00082]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart
BSW Parameter	BSW Type
IpduMJitUpdate	EcucBooleanParamDef
BSW Description	
If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00167]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter	BSW Type	
IpduMTxStaticConfirmation	EcucBooleanParamDef	
BSW Description		
A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the static part is generated.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00164]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter	BSW Type	
IpduMTxStaticHandleId	EcucIntegerParamDef	
BSW Description		
This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured static segments are copied and the IpduMTxTriggerMode is honored.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00129]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter	BSW Type	
IpduMTxStaticPduRef	EcucReferenceDef	
BSW Description		
Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00128]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	

BSW Parameter		BSW Type
IpduMTxStaticSegment		EcucParamConfContainerDef
BSW Description		
<p>The static part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxStaticSegment container shall be created that contains the location and the length of the segment.</p> <p>Please note that each segment in the source I-Pdu that is referenced in the IpduMTxStaticPart container will be copied to the same location in the multiplexed outgoing I-Pdu.</p>		
Template Description		
<p>The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.</p> <p>The ISignalPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalPdu are copied into this first segment and so on.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		Mapping Type
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00171]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSegment	
BSW Parameter		BSW Type
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
Template Description		
Data Length of the segment in bits.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00114]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSegment	
BSW Parameter		BSW Type
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
Template Description		

Segments bit position relatively to the beginning of a multiplexed IPdu.

Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. 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.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition

Mapping Rule

1:1 mapping

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	BSW Type
IpduMTxTriggerMode	EcucEnumerationParamDef
BSW Description	
Selects whether to send the multiplexed I-PDU immediately or at some later date.	
Template Description	
IPduM can be configured to send a transmission request for the new multiplexed IPdu to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration.	
In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MultiplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.triggerMode	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00125]

BSW Module	BSW Context
IpduM	IpduM
BSW Parameter	BSW Type
IpduMGeneral	EcucParamConfContainerDef
BSW Description	
Contains the general configuration parameters of IpduM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00130]

BSW Module		BSW Context	
IpduM		IpduM/IpduMGeneral	
BSW Parameter		BSW Type	
IpduMContainedTxPduPriorityHandling		EcucBooleanParamDef	
BSW Description			
This parameter enables/disables handling of priority for IpduMContainedTxPdu's with IpduMContainedTxPduCollectionSemantics IPDUM_LAST_IS_BEST. true: enabled false: disabled			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_IpduM_00209]	

BSW Module		BSW Context	
IpduM		IpduM/IpduMGeneral	
BSW Parameter		BSW Type	
IpduMDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_IpduM_00132]	

BSW Module		BSW Context	
IpduM		IpduM/IpduMGeneral	
BSW Parameter		BSW Type	
IpduMHeaderByteOrder		EcucEnumerationParamDef	
BSW Description			
This parameter defines the ByteOrder of the headers inside a Container I-PDU.			
Template Description			
Defines the byteOrder of the header in ContainerIPdus.			
M2 Parameter			
SystemTemplate::System.containerIPduHeaderByteOrder			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_IpduM_00197]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral/IpduMHeaderByteOrder	
BSW Parameter	BSW Type	
IPDUM_BIG_ENDIAN	EcucEnumerationLiteralDef	
BSW Description	Headers inside a Container I-PDU shall be ordered big endian.	
Template Description	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format)	
M2 Parameter	GenericStructure::GeneralTemplateClasses::PrimitiveTypes::ByteOrderEnum.mostSignificantByte First	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral/IpduMHeaderByteOrder	
BSW Parameter	BSW Type	
IPDUM_LITTLE_ENDIAN	EcucEnumerationLiteralDef	
BSW Description	Headers inside a Container I-PDU shall be ordered little endian.	
Template Description	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format)	
M2 Parameter	GenericStructure::GeneralTemplateClasses::PrimitiveTypes::ByteOrderEnum.mostSignificantByte Last	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral	
BSW Parameter	BSW Type	
IpduMMetaDataSupport	EcucBooleanParamDef	
BSW Description	This parameter enables/disables the support of meta-data feature. true: enabled false: disabled	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00205]	

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral	
BSW Parameter		BSW Type
IpduMStaticPartExists		EcucBooleanParamDef
BSW Description		
This is to allow optimizations in the case the IpduM will never be used with a static part. Note that this is a pre-compile option. If this is set to False then it will not be possible to add static parts after compilation. True: A static part may exist. False: A static part will never exist.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00133]

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral	
BSW Parameter		BSW Type
IpduMVersionInfoApi		EcucBooleanParamDef
BSW Description		
Active/Deactivate the version information API. true: version information activated false: version information deactivated		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_IpduM_00134]

BSW Module	BSW Context	
IpduM	IpduM	
BSW Parameter		BSW Type
IpduMPublishedInformation		EcucParamConfContainerDef
BSW Description		
Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_IpduM_00141]

BSW Module	BSW Context	
IpduM	IpduM/IpduMPublishedInformation	
BSW Parameter	BSW Type	
IpduMRxDirectComInvocation	EcucBooleanParamDef	
BSW Description	If set to TRUE the COM invocation optimization as defined in IPDUM140 is implemented.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_IpduM_00142]	

C.1.4 SecOc Mapping

BSW Module	BSW Context	
SecOC	SecOC	
BSW Parameter	BSW Type	
SecOCGeneral	EcucParamConfContainerDef	
BSW Description	Contains the general configuration parameters of the SecOC module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00002]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCGeneral	
BSW Parameter	BSW Type	
SecOCDefaultAuthenticationInformationPattern	EcucIntegerParamDef	
BSW Description	<p>The parameter describes the behaviour of SecOC when authentication build counter has reached the configuration value SecOCAuthenticationBuildAttempts, or the query of the freshness function returns E_NOT_OK or the calculation of the authenticator has returned a non-recoverable error such as returning E_NOT_OK or KEY_FAILURE.</p> <p>If the configuration parameter is not present, SecOC module shall remove the Authentic I-PDU from its internal buffer and cancel the transmission request</p> <p>If the configuration parameter is present, SecOC will use this value for each byte of Freshness Value and Authenticator when building the Authentication Information, and will not cancel the transmission request.</p>	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00098]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00007]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCEcucPartitionRef	EcucReferenceDef
BSW Description	
Reference to EcucPartition, where SecOC module is assigned to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00100]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCEnableForcedPassOverride	EcucBooleanParamDef
BSW Description	
When this configuration option is set to TRUE then the functionality inside the function SecOC_VerifyStatusOverride to send I-PDUs to upper layer independent of the verification result is enabled.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00051]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCIgnoreVerificationResult	EcucBooleanParamDef
BSW Description	
The result of the authentication process (e.g. MAC Verify) is ignored after the first try and the SecOC proceeds like the result was a success. The calculation of the authenticator is still done, only its result will be ignored. - true: enabled (verification result is ignored). - false: disabled (verification result is NOT ignored).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00052]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCMainFunctionPeriodRx	EcucFloatParamDef
BSW Description	
Allows to configure the time for the MainFunction of the Rx path (as float in seconds).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00053]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCMainFunctionPeriodTx	EcucFloatParamDef
BSW Description	
Allows to configure the time for the MainFunction of the Tx path (as float in seconds).	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00054]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCMaxAlignScalarType	EcucStringParamDef
BSW Description	
The scalar type which has the maximum alignment restrictions on the given platform. This type can be e.g. uint8, uint16 or uint32.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00047]

BSW Module	BSW Context
SecOC	SecOC/SecOCGeneral
BSW Parameter	BSW Type
SecOCOverrideStatusWithDataId	EcucBooleanParamDef
BSW Description	
This option defines if the parameter "ValueId" of the function SecOC_VerifyStatusOverride() accepts the freshness value (as a collection of one or more Secured I-PDUs to freshness) or the dataId for individual Secured I-PDUs. * true: Function SecOC_VerifyStatusOverride accepts SecOCDataId as parameter. * false: Function SecOC_VerifyStatusOverride accepts SecOCFreshnessValueId as parameter.	
Template Description	
ContainedIPduProps.headerIdLongHeader: Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.	
ContainedIPduProps.headerIdShortHeader: Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdLongHeader, SystemTemplate::Fibex::FibexCore::CoreCommunication::ContainedIPduProps.headerIdShortHeader	
Mapping Rule	Mapping Type
If IpduMContainerHeaderSize = LONG the IPduMContainedPduHeaderId is taken from headerIdLongHeader. If IpduMContainerHeaderSize = SHORT the IPduMContainedPduHeaderId is taken from headerIdShortHeader.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00099]

BSW Module		BSW Context	
SecOC		SecOC/SecOCGeneral	
BSW Parameter		BSW Type	
SecOCQueryFreshnessValue		EcucEnumerationParamDef	
BSW Description			
This parameter specifies if the freshness value shall be determined through a C-function (CD) or a software component (SW-C).			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SecOC_00078]	

BSW Module		BSW Context	
SecOC		SecOC/SecOCGeneral	
BSW Parameter		BSW Type	
SecOCVerificationStatusCallout		EcucFunctionNameDef	
BSW Description			
Entry address of the customer specific call out routine which shall be invoked in case of a verification attempt.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SecOC_00004]	

BSW Module		BSW Context	
SecOC		SecOC/SecOCGeneral	
BSW Parameter		BSW Type	
SecOCVersionInfoApi		EcucBooleanParamDef	
BSW Description			
If true the SecOC_GetVersionInfo API is available.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SecOC_00003]	

BSW Module		BSW Context	
SecOC		SecOC	
BSW Parameter		BSW Type	

SecOCRxPduProcessing	EcucParamConfContainerDef
BSW Description	
Contains the parameters to configure the RxPdus to be verified by the SecOC module.	
Template Description	
If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator).	
If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu	
Mapping Rule	Mapping Type
This container shall be created for every SecuredIPdu that is received by the regarded Ecu. The information whether the SecuredIPdu is transmitted or received by the Ecu shall be derived from PduTriggering.iPduPort reference. If an IPduPort of the Ecu with the communicationDirection = out is referenced then the SecuredIPdu is transmitted. If an IPduPort of the Ecu with the communicationDirection = in is referenced then the SecuredIPdu is received.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00011]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCAuthDataFreshnessLen	EcucIntegerParamDef
BSW Description	
The length of the external authentic PDU data in bits (uint16).	
Template Description	
This attribute defines the length in bits of the authentic PDU data that is passed to the SWC that verifies and generates the Freshness.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.authDataFreshnessLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00082]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCAuthDataFreshnessStartPosition	EcucIntegerParamDef
BSW Description	
This value determines the start position in bits (uint16) of the Authentic PDU that shall be passed on to the Freshness SWC. The bit position starts counting from the MSB of the first byte of the PDU.	
Template Description	
This value determines the start position in bits of the Authentic PDU that shall be passed on to the SWC that verifies and generates the Freshness.. The bit position starts counting from the MSB of the first byte of the PDU.	

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.authDataFreshnessStartPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00081]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCAuthInfoTruncLength	EcucIntegerParamDef
BSW Description	
This parameter defines the length in bits of the authentication code to be included in the payload of the Secured I-PDU.	
Template Description	
This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationAuthenticationProps.authInfoTxLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCAuthenticationBuildAttempts	EcucIntegerParamDef
BSW Description	
This parameter specifies the number of authentication build attempts.	
Template Description	
This attribute specifies the number of authentication build attempts.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.authenticationBuildAttempts	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00079]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCAuthenticationVerifyAttempts	EcucIntegerParamDef
BSW Description	
This parameter specifies the number of authentication verify attempts that are to be carried out when the verification of the authentication information failed for a given Secured I-PDU. If zero is set, then only one authentication verification attempt is done.	

Template Description	
This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given SecuredIPdu. If zero is set than only one authentication attempt is done.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunication Props.authenticationRetries	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00080]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCDataId	EcucIntegerParamDef
BSW Description	
This parameter defines a unique numerical identifier for the Secured I-PDU.	
Template Description	
This attribute defines a numerical identifier for the Secured I-PDU.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.dataId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCFreshnessValueId	EcucIntegerParamDef
BSW Description	
This parameter defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
Template Description	
This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.freshnessValueId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCFreshnessValueLength	EcucIntegerParamDef
BSW Description	

This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
Template Description	
This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationFreshnessProps.freshnessValueLength	
Mapping Rule	Mapping Type
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCFreshnessValueTruncLength	EcucIntegerParamDef
BSW Description	
This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured I-PDU.	
Template Description	
This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationFreshnessProps.freshnessValueTxLength	
Mapping Rule	Mapping Type
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueTxLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCReceptionOverflowStrategy	EcucEnumerationParamDef
BSW Description	
This parameter defines the overflow strategy for receiving PDUs	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00076]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter	BSW Type	
SecOCReceptionQueueSize	EcucIntegerParamDef	
BSW Description		
This parameter defines the queue size in case the overflow strategy for receiving PDUs is set to QUEUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00077]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter	BSW Type	
SecOCRxAuthServiceConfigRef	EcucReferenceDef	
BSW Description		
This reference is used to define which crypto service function is called for authentication. If PDUs with a dynamic length are used (e.g. CanTP or Dynamic Length PDUs) a MAC algorithm has to be chosen, that is not vulnerable to length extension attack (e.g. CMAC/HMAC).		
Template Description		
This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced SecuredIPdu.		
Obviously, this reference is only applicable if the PduTriggering also references a SecuredIPdu in the role iPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.secOcCryptoMapping		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00048]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter	BSW Type	
SecOCRxAuthenticPduLayer	EcucParamConfContainerDef	
BSW Description		
This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was verified.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00044]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxAuthenticPduLayer
BSW Parameter	BSW Type
SecOCPduType	EcucEnumerationParamDef
BSW Description	
This parameter defines API Type to use for communication with PduR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00075]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxAuthenticPduLayer
BSW Parameter	BSW Type
SecOCRxAuthenticLayerPduRef	EcucReferenceDef
BSW Description	
Reference to the global Pdu.	
Template Description	
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.payload	
Mapping Rule	Mapping Type
Reference to the EcuC Pdu which represents the verified SecuredIPdu Pdu Triggering	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00045]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCRxPduSecuredArea	EcucParamConfContainerDef
BSW Description	
This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator verification algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator verification algorithm.	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	

SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaOffset	
Mapping Rule	Mapping Type
Create container if the securedAreaOffset and securedAreaLength is defined for the SecuredIPdu in the System Description.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00089]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxPduSecuredArea
BSW Parameter	BSW Type
SecOCSecuredRxPduLength	EcucIntegerParamDef
BSW Description	
This parameter defines the length (in bytes) of the area within the Pdu which is secured	
Template Description	
This attribute defines the length in bytes of the area within the payload Pdu which will be secured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00091]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxPduSecuredArea
BSW Parameter	BSW Type
SecOCSecuredRxPduOffset	EcucIntegerParamDef
BSW Description	
This parameter defines the start position (offset in bytes) of the area within the Pdu which is secured	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00090]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCRxSecuredPduLayer	EcucChoiceContainerDef
BSW Description	
This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac verification is provided.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00041]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer
BSW Parameter	BSW Type
SecOCRxSecuredPdu	EcucParamConfContainerDef
BSW Description	
This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac verification is provided.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.	
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCRxSecuredPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = false or not defined	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00069]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu
BSW Parameter	BSW Type
SecOCAuthPduHeaderLength	EcucIntegerParamDef
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	
This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useSecuredPduHeader	
Mapping Rule	Mapping Type

<p>The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecuredIPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "securedPduHeader32Bit" the value shall be 4.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00093]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu	
BSW Parameter	BSW Type	
SecOCRxSecuredLayerPduId	EcucIntegerParamDef	
BSW Description		
PDU identifier assigned by SecOC module. Used by PduR for <nowiki>SecOC_[f T]RxIndication</nowiki>.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00043]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu	
BSW Parameter	BSW Type	
SecOCRxSecuredLayerPduRef	EcucReferenceDef	
BSW Description		
Reference to the global Pdu.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu		
Mapping Rule	Mapping Type	
Reference to the EcuC Pdu which represents the received SecuredIPdu Pdu Triggering	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00042]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu	
BSW Parameter	BSW Type	
SecOCSecuredRxPduVerification	EcucBooleanParamDef	

BSW Description	
This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.	
Template Description	
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduPort.rxSecurityVerification	
Mapping Rule	Mapping Type
SecOCSecuredRxPduVerification is true if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00092]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer
BSW Parameter	BSW Type
SecOCRxSecuredPduCollection	EcucParamConfContainerDef
BSW Description	
This container specifies two Pdus that are received by the SecOC module from the PduR and a message linking between them. SecOCRxAuthenticPdu contains the original Authentic I-PDU, i.e. the secured data, and the SecOCRxCryptographicPdu contains the Authenticator, i.e. the actual Authentication Information.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCRxSecuredPduCollection if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00067]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection
BSW Parameter	BSW Type
SecOCRxAuthenticPdu	EcucParamConfContainerDef
BSW Description	
This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.	

Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.	
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCRxAutenticPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00061]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAutenticPdu
BSW Parameter	BSW Type
SecOCAuthPduHeaderLength	EcucIntegerParamDef
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	
This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useSecuredPduHeader	
Mapping Rule	Mapping Type
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecuredIPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "securedPduHeader32Bit" the value shall be 4.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00093]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAutenticPdu
BSW Parameter	BSW Type
SecOCRxAutenticPduId	EcucIntegerParamDef
BSW Description	
PDU identifier of the Authentic I-PDU assigned by SecOC module. Used by PduR for SecOC_IfRxIndication.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00062]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxAuthenticPdu
BSW Parameter	BSW Type
SecOCRxAuthenticPduRef	EcucReferenceDef
BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu	
Mapping Rule	Mapping Type
Reference to the EcuC Pdu which represents the received Authentic Secured Pdu PduTriggering referenced by SecuredIPdu.payload	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00063]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection
BSW Parameter	BSW Type
SecOCRxCryptographicPdu	EcucParamConfContainerDef
BSW Description	
This container specifies the Cryptographic Pdu that is received by the SecOC module from the PduR.	
Template Description	
<p>If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.</p> <p>If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCRxCryptographicPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true and SecuredIPdu.payload refers to a PduTriggering which the Ecu Instance receives as well.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00064]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxCryptographicPdu
BSW Parameter	BSW Type

SecOCRxCryptographicPduId	EcucIntegerParamDef
BSW Description	
PDU identifier of the Cryptographic I-PDU assigned by SecOC module. Used by PduR for SecOC_IfRxIndication.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00065]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxCryptographicPdu	
BSW Parameter	BSW Type	
SecOCRxCryptographicPduRef	EcucReferenceDef	
BSW Description		
Reference to the global Pdu.		
Template Description		
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.payload		
Mapping Rule		Mapping Type
Reference to the EcuC Pdu which represents the received Cryptographic SecuredIPdu PduTriggering		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00066]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection	
BSW Parameter	BSW Type	
SecOCSecuredRxPduVerification	EcucBooleanParamDef	
BSW Description		
This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.		
Template Description		
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduPort.rxSecurityVerification		
Mapping Rule		Mapping Type
SecOCSecuredRxPduVerification is true if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification.		full
Mapping Status	ECUC Parameter ID	

valid	[ECUC_SecOC_00092]
-------	--------------------

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection	
BSW Parameter		BSW Type
SecOCUseMessageLink		EcucParamConfContainerDef
BSW Description		
SecOC links an Authentic I-PDU and Cryptographic I-PDU together by repeating a specific part (Message Linker) of the Authentic I-PDU in the Cryptographic I-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00074]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCUseMessageLink	
BSW Parameter		BSW Type
SecOCMessageLinkLen		EcucIntegerParamDef
BSW Description		
Length of the Message Linker inside the Authentic I-PDU in bits.		
Template Description		
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.messageLinkLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00060]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCUseMessageLink	
BSW Parameter		BSW Type
SecOCMessageLinkPos		EcucIntegerParamDef
BSW Description		
The position of the Message Linker inside the Authentic I-PDU in bits.		
Template Description		
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.		
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.messageLinkPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00059]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCSameBufferPduRef	EcucReferenceDef
BSW Description	
This reference is used to collect Pdus that are using the same SecOC buffer.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCUseAuthDataFreshness	EcucBooleanParamDef
BSW Description	
A Boolean value that indicates if a part of the Authentic-PDU shall be passed on to the SWC that verifies and generates the Freshness. If it is set to TRUE, the values SecOCAuthDataFreshnessStartPosition and SecOCAuthDataFreshnessLen must be set to specify the bit position and length within the Authentic-PDU.	
Template Description	
This attribute describes whether a part of AuthenticPdu contained in a SecuredIPdu shall be passed on to the SWC that verifies and generates the Freshness. The part of the Authentic-PDU is defined by the authDataFreshnessStartPosition and authDataFreshnessLength.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduPort.useAuthDataFreshness	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00083]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	BSW Type
SecOCVerificationStatusPropagationMode	EcucEnumerationParamDef
BSW Description	
This parameter is used to describe the propagation of the status of each verification attempt from the SecOC module to SWCs.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00046]

BSW Module	BSW Context
SecOC	SecOC
BSW Parameter	BSW Type
SecOCSameBufferPduCollection	EcucParamConfContainerDef
BSW Description	
SecOCBuffer configuration that may be used by a collection of Pdus.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00009]

BSW Module	BSW Context
SecOC	SecOC/SecOCSameBufferPduCollection
BSW Parameter	BSW Type
SecOCBufferLength	EcucIntegerParamDef
BSW Description	
This parameter defines the Buffer in bytes that is used by the SecOC module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00008]

BSW Module	BSW Context
SecOC	SecOC
BSW Parameter	BSW Type
SecOCTxPduProcessing	EcucParamConfContainerDef
BSW Description	
Contains the parameters to configure the TxPdus to be secured by the SecOC module.	
Template Description	

<p>If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator).</p> <p>If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu	
Mapping Rule	Mapping Type
<p>This container shall be created for every SecuredIPdu that is transmitted by the regarded Ecu.</p> <p>The information whether the SecuredIPdu is transmitted or received by the Ecu shall be derived from PduTriggering.iPduPort reference. If an IPduPort of the Ecu with the communicationDirection = out is referenced then the SecuredIPdu is transmitted. If an IPduPort of the Ecu with the communicationDirection = in is referenced then the SecuredIPdu is received.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00012]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCAuthInfoTruncLength	EcucIntegerParamDef
BSW Description	
This parameter defines the length in bits of the authentication code to be included in the payload of the Secured I-PDU.	
Template Description	
This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationAuthentication Props.authInfoTxLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCAuthenticationBuildAttempts	EcucIntegerParamDef
BSW Description	
This parameter specifies the number of authentication build attempts.	
Template Description	
This attribute specifies the number of authentication build attempts.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunication Props.authenticationBuildAttempts	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_SecOC_00079]
-------	--------------------

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCDataId	EcucIntegerParamDef
BSW Description	
This parameter defines a unique numerical identifier for the Secured I-PDU.	
Template Description	
This attribute defines a numerical identifier for the Secured I-PDU.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.dataId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCFreshnessValueId	EcucIntegerParamDef
BSW Description	
This parameter defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
Template Description	
This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.freshnessValueId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCFreshnessValueLength	EcucIntegerParamDef
BSW Description	
This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
Template Description	
This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationFreshnessProps.freshnessValueLength	

Mapping Rule		Mapping Type
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshness ValueLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCFreshnessValueTruncLength		EcucIntegerParamDef
BSW Description		
This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured I-PDU.		
Template Description		
This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationFreshness Props.freshnessValueTxLength		
Mapping Rule		Mapping Type
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshness ValueTxLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCProvideTxTruncatedFreshnessValue		EcucBooleanParamDef
BSW Description		
This parameter specifies if the Tx query freshness function provides the truncated freshness info instead of generating this by SecOC In this case, SecOC shall add this data to the Authentic PDU instead of truncating the freshness value.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00084]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type

SecOCSameBufferPduRef	EcucReferenceDef
BSW Description	
This reference is used to collect Pdus that are using the same SecOC buffer.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCTxAuthServiceConfigRef		EcucReferenceDef
BSW Description		
This reference is used to define which crypto service function is called for authentication. If PDUs with a dynamic length are used (e.g. CanTP or Dynamic Length PDUs) a MAC algorithm has to be chosen, that is not vulnerable to length extension attack (e.g. CMAC/HMAC).		
Template Description		
This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced SecuredIPdu.		
Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role iPdu.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.secOcCryptoMapping		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00013]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCTxAuthenticPduLayer		EcucParamConfContainerDef
BSW Description		
This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac generation is provided.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00023]	

BSW Module	BSW Context
-------------------	--------------------

SecOC	SecOC/SecOCTxPduProcessing/SecOCTxAuthenticPduLayer	
BSW Parameter		BSW Type
SecOCPduType	EcucEnumerationParamDef	
BSW Description		
This parameter defines API Type to use for communication with PduR.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00075]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxAuthenticPduLayer	
BSW Parameter		BSW Type
SecOCTxAuthenticLayerPduId	EcucIntegerParamDef	
BSW Description		
PDU identifier assigned by SecOC module. Used by PduR for <nowiki>SecOC_[If Tp]Transmit</nowiki>.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00026]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxAuthenticPduLayer	
BSW Parameter		BSW Type
SecOCTxAuthenticLayerPduRef	EcucReferenceDef	
BSW Description		
Reference to the global Pdu.		
Template Description		
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.payload		
Mapping Rule		Mapping Type
Reference to the EcuC Pdu which represents the to be secured SecuredIPdu PduTriggering		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00025]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCTxPduSecuredArea	EcucParamConfContainerDef	

BSW Description	
This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator generation algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator generation algorithm.	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaOffset	
Mapping Rule	Mapping Type
Create container if the securedAreaOffset and securedAreaLength is defined for the SecuredIPdu in the System Description.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00086]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxPduSecuredArea
BSW Parameter	BSW Type
SecOCSecuredTxPduLength	EcucIntegerParamDef
BSW Description	
This parameter defines the length (in bytes) of the area within the Pdu which shall be secured	
Template Description	
This attribute defines the length in bytes of the area within the payload Pdu which will be secured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00088]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxPduSecuredArea
BSW Parameter	BSW Type
SecOCSecuredTxPduOffset	EcucIntegerParamDef
BSW Description	
This parameter defines the start position (offset in bytes) of the area within the Pdu which shall be secured	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.securedAreaOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00087]

BSW Module	BSW Context
-------------------	--------------------

SecOC	SecOC/SecOCTxPduProcessing	
BSW Parameter		BSW Type
SecOCTxSecuredPduLayer		EcucChoiceContainerDef
BSW Description		
This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00024]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer	
BSW Parameter		BSW Type
SecOCTxSecuredPdu		EcucParamConfContainerDef
BSW Description		
This container specifies one Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. This Pdu contains the cryptographic information.		
Template Description		
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.		
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu		
Mapping Rule		Mapping Type
Create the SecOCTxSecuredPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = false or not defined		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00070]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu	
BSW Parameter		BSW Type
SecOCAuthPduHeaderLength		EcucIntegerParamDef
BSW Description		
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.		
Template Description		
This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useSecuredPduHeader		
Mapping Rule		Mapping Type

<p>The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecuredIPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "securedPduHeader32Bit" the value shall be 4.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00093]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu	
BSW Parameter	BSW Type	
SecOCTxSecuredLayerPduId	EcucIntegerParamDef	
BSW Description		
PDU identifier assigned by SecOC module. Used by PduR for confirmation (<nowiki>SecOC_[If Tp]TxConfirmation</nowiki>) and for Trigger-Transmit.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00028]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu	
BSW Parameter	BSW Type	
SecOCTxSecuredLayerPduRef	EcucReferenceDef	
BSW Description		
Reference to the global Pdu.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu		
Mapping Rule	Mapping Type	
Reference to the EcuC Pdu which represents the sent SecuredIPdu PduTriggering	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00027]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer	
BSW Parameter	BSW Type	
SecOCTxSecuredPduCollection	EcucParamConfContainerDef	

BSW Description	
This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. Two separate Pdus are transmitted to the PduR: Authentic I-PDU and Cryptographic I-PDU.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.	
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCTxSecuredPduCollection if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00071]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection
BSW Parameter	BSW Type
SecOCTxAuthenticPdu	EcucParamConfContainerDef
BSW Description	
This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.	
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	Mapping Type
Create the SecOCTxAuthenticPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00072]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu
BSW Parameter	BSW Type
SecOCAuthPduHeaderLength	EcucIntegerParamDef
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	

This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useSecuredPduHeader	
Mapping Rule	Mapping Type
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecuredIPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "securedPduHeader32Bit" the value shall be 4.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00093]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu
BSW Parameter	BSW Type
SecOCTxAuthenticPduId	EcucIntegerParamDef
BSW Description	
PDU identifier of the Authentic I-PDU assigned by SecOC module. Used by PduR for confirmation (SecOC_IftxConfirmation) and for TriggerTransmit.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00055]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu
BSW Parameter	BSW Type
SecOCTxAuthenticPduRef	EcucReferenceDef
BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu	
Mapping Rule	Mapping Type
Reference to the Ecuc Pdu which represents the sent Authentic SecuredIPdu PduTriggering referenced by SecuredIPdu.payload	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00056]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection	
BSW Parameter		BSW Type
SecOCTxCryptographicPdu		EcucParamConfContainerDef
BSW Description		
This container specifies the Cryptographic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.		
Template Description		
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.		
If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.useAsCryptographicIPdu		
Mapping Rule		Mapping Type
Create the SecOCTxCryptographicPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true and SecuredIPdu.payload refers to a PduTriggering which the Ecu Instance sends as well.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00073]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxCryptographicPdu	
BSW Parameter		BSW Type
SecOCTxCryptographicPduId		EcucIntegerParamDef
BSW Description		
PDU identifier of the Cryptographic I-PDU assigned by SecOC module. Used by PduR for confirmation (SecOC_IfTxConfirmation) and for TriggerTransmit.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SecOC_00057]

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxCryptographicPdu	
BSW Parameter		BSW Type
SecOCTxCryptographicPduRef		EcucReferenceDef
BSW Description		
Reference to the global Pdu.		
Template Description		
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecuredIPdu.payload		

Mapping Rule		Mapping Type
Reference to the EcuC Pdu which represents the sent Cryptographic SecuredI Pdu PduTriggering		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00058]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection	
BSW Parameter	BSW Type	
SecOCUseMessageLink	EcucParamConfContainerDef	
BSW Description		
SecOC links an Authentic I-PDU and Cryptographic I-PDU together by repeating a specific part (Message Linker) of the Authentic I-PDU in the Cryptographic I-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00074]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCUseMessageLink	
BSW Parameter	BSW Type	
SecOCMessageLinkLen	EcucIntegerParamDef	
BSW Description		
Length of the Message Linker inside the Authentic I-PDU in bits.		
Template Description		
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.messageLinkLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SecOC_00060]	

BSW Module	BSW Context	
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCUseMessageLink	
BSW Parameter	BSW Type	
SecOCMessageLinkPos	EcucIntegerParamDef	
BSW Description		
The position of the Message Linker inside the Authentic I-PDU in bits.		
Template Description		

SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SecureCommunicationProps.messageLinkPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00059]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	BSW Type
SecOCUseTxConfirmation	EcucBooleanParamDef
BSW Description	
A Boolean value that indicates if the function SecOC_SPduTxConfirmation shall be called for this PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00085]

C.1.5 PduR

BSW Module	BSW Context
PduR	PduR
BSW Parameter	BSW Type
PduRBswModules	EcucParamConfContainerDef
BSW Description	
Each container describes a specific BSW module (upper/CDD/lower/lpduM) that the PDU Router shall interface to.	
The reason to have it as own configuration container instead of implication of the routing path is to be able to configure CDDs properly and to force module's to be used in a post-build situation even though no routing is made to/from this module (future configurations may include these modules).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00295]

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter	BSW Type	
PduRBswModuleRef	EcucForeignReferenceDef	
BSW Description		
<p>This is a reference to one BSW module's configuration (i.e. not the ECUC parameter definition template).</p> <p>Example, there could be several configurations of LinIf and this reference selects one of them.</p>		
Template Description		
<p>Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure.</p> <p>As part of the BSW module description, the EcucModuleConfigurationValues element has two different roles:</p> <p>The recommendedConfiguration contains parameter values recommended by the BSW module vendor.</p> <p>The preconfiguredConfiguration contains values for those parameters which are fixed by the implementation and cannot be changed.</p> <p>These two EcucModuleConfigurationValues are used when the base EcucModuleConfigurationValues (as part of the base ECU configuration) is created to fill parameters with initial values.</p>		
M2 Parameter		
ECUCDescriptionTemplate::EcucModuleConfigurationValues		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00294]

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter	BSW Type	
PduRCancelReceive	EcucBooleanParamDef	
BSW Description		
<p>Specifies if the Transport protocol module supports the CancelReceive API or not. Value true the API is supported.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00340]

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter	BSW Type	
PduRCancelTransmit	EcucBooleanParamDef	
BSW Description		

Specifies if the BSW module supports the CancelTransmit API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00297]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRCommunicationInterface	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the Communication Interface APIs or not. Value true the APIs are supported.	
A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the COM module).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00298]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRCopyRxData	EcucBooleanParamDef
BSW Description	
Specifies if the Transport protocol module supports the CopyRxData API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00360]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRCopyTxData	EcucBooleanParamDef

BSW Description	
Specifies if the Transport protocol module supports the CopyTxData API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00362]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRLowerModule	EcucBooleanParamDef
BSW Description	
<p>The PduRLowerModule will decide who will call the APIs and who will implement the APIs.</p> <p>For example, if the CanIf module is referenced then the PDU Router module will implement the PduR_CanIfRxIndication API. And the PDUR module will call the CanIf_Transmit API. Other APIs are of course also covered.</p> <p>An upper module can also be an lower module (e.g. the IpduM module).</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00307]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRRetransmission	EcucBooleanParamDef
BSW Description	
<p>If set to true this means that the destination transport protocol module will use the retransmission feature. This parameter might be set to false if the retransmission feature is not used, even though the destination transport protocol is supporting it.</p> <p>This parameter is only valid for transport protocol modules and gateway operations. If transmission from a local upper layer module this module will handle the retransmission.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00332]

BSW Module		BSW Context	
PduR		PduR/PduRBswModules	
BSW Parameter		BSW Type	
PduRRxIndication		EcucBooleanParamDef	
BSW Description			
Specifies if BSW module supports the RxIndication API or not. Value true the API is supported.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00358]	

BSW Module		BSW Context	
PduR		PduR/PduRBswModules	
BSW Parameter		BSW Type	
PduRStartOfReception		EcucBooleanParamDef	
BSW Description			
Specifies if the Transport protocol module supports the StartOfReception API or not. Value true the API is supported.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00359]	

BSW Module		BSW Context	
PduR		PduR/PduRBswModules	
BSW Parameter		BSW Type	
PduRTpRxIndication		EcucBooleanParamDef	
BSW Description			
Specifies if the Transport protocol module supports the TpRxIndication API or not. Value true the API is supported.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00364]	

BSW Module		BSW Context	
PduR		PduR/PduRBswModules	
BSW Parameter		BSW Type	

PduRTpTransmit	EcucBooleanParamDef
BSW Description	
Specifies if BSW module supports the TP Transmit API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00361]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRTpTxConfirmation	EcucBooleanParamDef
BSW Description	
Specifies if the Transport protocol module supports the TpTxConfirmation API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00363]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRTransmit	EcucBooleanParamDef
BSW Description	
Specifies if BSW module supports the (IF) Transmit API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00357]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRTransportProtocol	EcucBooleanParamDef
BSW Description	
The PDU Router module shall use the API parameters specified for transport protocol interface.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00312]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRTriggertransmit	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the TriggerTransmit API or not. Value true means that the BSW module supports the TriggerTransmit interface which a lower layer module can call and also that it can call the TriggerTransmit interface of an upper layer module. Value false means that the BSW module does not support the TriggerTransmit interface which a lower layer module can call and also that it shall not call the TriggerTransmit interface of an upper layer module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00313]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRTxConfirmation	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the TxConfirmation API or not. Value true the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00314]

BSW Module	BSW Context
PduR	PduR/PduRBswModules
BSW Parameter	BSW Type
PduRUpperModule	EcucBooleanParamDef
BSW Description	

The PduRUpperModule will decide who will call the APIs and who will implement the APIs.

For example, if the COM module is referenced then the PDU Router module will implement the PduR_Transmit API. And the PDUR module will call the Com_RxIndication API. Other APIs are of course also covered.

An upper module can also be a lower module (e.g. the IpduM module).

Template Description

M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

local

Mapping Status	ECUC Parameter ID
----------------	-------------------

valid	[ECUC_PduR_00338]
-------	-------------------

BSW Module	BSW Context
------------	-------------

PduR	PduR/PduRBswModules
------	---------------------

BSW Parameter	BSW Type
---------------	----------

PduRUseTag	EcucBooleanParamDef
------------	---------------------

BSW Description

This parameter, if set to true, enables the usage of the tag (<up>) in the following API calls:

- * PduR_<Up>CancelReceive
- * PduR_<Up>CancelTransmit

Example: If used by COM and the parameter is enabled the PduR_ComCancelTransmit is used.

The background is that upper layer modules differ in usage of this tag (e.g. COM is using the tag, DCM is not).

Template Description

M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

local

Mapping Status	ECUC Parameter ID
----------------	-------------------

valid	[ECUC_PduR_00319]
-------	-------------------

BSW Module	BSW Context
------------	-------------

PduR	PduR
------	------

BSW Parameter	BSW Type
---------------	----------

PduRGeneral	EcucParamConfContainerDef
-------------	---------------------------

BSW Description

This container is a subcontainer of PduR and specifies the general configuration parameters of the PDU Router.

Template Description

M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

local

Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00305]

BSW Module	BSW Context	
PduR	PduR/PduRGeneral	
BSW Parameter		BSW Type
PduRDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00302]	

BSW Module	BSW Context	
PduR	PduR/PduRGeneral	
BSW Parameter		BSW Type
PduRMetaDataSupport		EcucBooleanParamDef
BSW Description		
Enable support for MetaData handling. The MetaData is defined by the referenced MetaDataType of the global PDU definitions. This feature may be used for efficient address based routing and generic CAN-CAN-routing, where the MetaData contains the CAN ID.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00347]	

BSW Module	BSW Context	
PduR	PduR/PduRGeneral	
BSW Parameter		BSW Type
PduRVersionInfoApi		EcucBooleanParamDef
BSW Description		
If true the PduR_GetVersionInfo API is available.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_PduR_00316]
-------	-------------------

BSW Module	BSW Context	
PduR	PduR/PduRGeneral	
BSW Parameter		BSW Type
PduRZeroCostOperation		EcucBooleanParamDef
BSW Description		
If set the PduR configuration generator will report an error if zero-cost-operation cannot be fulfilled. This parameter shall be seen as an input requirement to the configuration generator.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00317]

BSW Module	BSW Context	
PduR	PduR	
BSW Parameter		BSW Type
PduRRoutingPaths		EcucParamConfContainerDef
BSW Description		
Represents one table of routing paths.		
This routing table allows multiple configurations that can be used to create several routing tables in the same configuration. This is mainly used for post-build (e.g. post-build selectable) but can be used by pre-compile and link-time for variant handling.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00310]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths	
BSW Parameter		BSW Type
PduRConfigurationId		EcucIntegerParamDef
BSW Description		
Identification of the configuration of the PduR configuration. This identification can be read using the PduR API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00327]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths	
BSW Parameter	BSW Type	
PduRDestPdu	EcucParamConfContainerDef	
BSW Description		
This container is a subcontainer of PduRRoutingPath and specifies one destination for the PDU to be routed.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00249]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRDestPdu	
BSW Parameter	BSW Type	
PduRDestPduDataProvision	EcucEnumerationParamDef	
BSW Description		
Specifies how data are provided: direct (as part of the Transmit call) or via the TriggerTransmit callback function. Only required for non-TP gatewayed I-PDUs.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00289]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRDestPdu	
BSW Parameter	BSW Type	
PduRDestPduHandleId	EcucIntegerParamDef	
BSW Description		
PDU identifier assigned by PDU Router. Used by communication interface and transport protocol modules for confirmation (PduR_<Lo>TxConfirmation) and for TriggerTransmit (PduR_<Lo>TriggerTransmit).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00322]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRDestPdu	
BSW Parameter	BSW Type	
PduRDestPduRef	EcucReferenceDef	
BSW Description		
Destination PDU reference; reference to unique PDU identifier which shall be used by the PDU Router instead of the source PDU ID when calling the related function of the destination module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00291]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRDestPdu	
BSW Parameter	BSW Type	
PduRTransmissionConfirmation	EcucBooleanParamDef	
BSW Description		
This parameter is only for communication interfaces. Transport protocol modules will always call the TxConfirmation function.		
If set the destination communication interface module will call the TxConfirmation. However the TxConfirmation may be not called due to error. So the PduR shall not block until the TxConfirmation is called.		
One background for this parameter is for the PduR to know when all modules have confirmed a multicast operation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00339]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths	
BSW Parameter	BSW Type	
PduRMaxRoutingPathCnt	EcucIntegerParamDef	
BSW Description		
Maximum number of RoutingPaths in all RoutingTables. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00350]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths
BSW Parameter	BSW Type
PduRMaxRoutingPathGroupCnt	EcucIntegerParamDef
BSW Description	
Maximum number of RoutingPathGroups. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00348]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths
BSW Parameter	BSW Type
PduRRoutingPath	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of PduRRoutingTable and specifies the routing path of a PDU.	
Template Description	
<p>IPduMapping: Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.</p> <p>PduTriggering: The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu.</p> <p>Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface.</p> <p>If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.</p> <p>TpConfig: Contains all configuration elements for AUTOSAR TP.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping, SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering, SystemTemplate::TransportProtocols::TpConfig	
Mapping Rule	Mapping Type

For each MultiplatformGateway.pduMapping; for each SignalPdu-Multiplexed Pdu Connection; for each IPduTriggering; for each TpConfig create one Pdu RRoutingPath.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00248]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRRoutingPath	
BSW Parameter	BSW Type	
PduRDefaultValue	EcucParamConfContainerDef	
BSW Description		
Specifies the default value of the I-PDU. Only required for gateway operation and if at least one PDU specified by PduRDestPdu uses TriggerTransmit Data provision.		
Represented as an array of IntegerParamDef.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::TargetIPduRef.defaultValue		
Mapping Rule		Mapping Type
Container should be created if PduMappingDefaultValue is described in the System		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00299]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue	
BSW Parameter	BSW Type	
PduRDefaultValueElement	EcucParamConfContainerDef	
BSW Description		
Each value element is represented by the element and the position in an array.		
Template Description		
The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement		
Mapping Rule		Mapping Type
Container shall be created for each DefaultValueElement that is aggregated by PduMappingDefaultValue.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00300]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue/PduRDefaultValueElement	
BSW Parameter	BSW Type	
PduRDefaultValueElement	EcucIntegerParamDef	
BSW Description		
The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength. The position of this parameter in the container is specified by the PduRElementBytePosition parameter.		

Template Description	
The integer value of a freely defined data byte.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement.elementByteValue	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00290]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue/PduRDefaultValueElement
BSW Parameter	BSW Type
PduRDefaultValueElementBytePosition	EcucIntegerParamDef
BSW Description	
This parameter specifies the byte position of the element within the default value	
Template Description	
This attribute specifies the byte position of the element within the default value	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement.elementPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00292]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	BSW Type
PduRDestPduRRef	EcucReferenceDef
BSW Description	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00354]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	BSW Type
PduRDestTxBufferRef	EcucReferenceDef
BSW Description	
Reference to a buffer in the PduR. This buffer is required for communication interface gatewaying, and for transport protocol gatewaying.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00304]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	BSW Type
PduRQueueDepth	EcucIntegerParamDef
BSW Description	
This parameter defines the queue depth for this routing path.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00356]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	BSW Type
PduRRoutingPathGroupRef	EcucReferenceDef
BSW Description	
Reference to routing path destinations.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduIPduGroup.iPdu	
Mapping Rule	Mapping Type
If the PduTriggering this PduRRoutingPath is derived from is the target of a Pdu IPduGroup.iPdu reference then a PduRRoutingPathGroupRef shall be created at the PduRRoutingPath and the respective PduRRoutingPathGroup shall be referenced from the PduRRoutingPathGroupRef.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00352]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	BSW Type
PduRSrcPduRRef	EcucReferenceDef
BSW Description	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00353]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRRoutingPath	
BSW Parameter		BSW Type
PduRTpThreshold		EcucIntegerParamDef
BSW Description		
<p>This parameter is only relevant for TP routings.</p> <p>When configured, it enables on-the-fly routing and defines the number of bytes which must have been received before transmission on the destination bus may start.</p> <p>When omitted, direct TP routing is enforced. The PduRouter shall ensure that a buffer is allocated for this routing path which is at least as large as the threshold.</p>		
Template Description		
Optionally defines the to be configured Pdu Router TpChunkSize for this routing relation.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping.pdurTpChunkSize		
Mapping Rule		Mapping Type
PduRTpThreshold shall only be configured by IpduMapping.pduRTpChunkSize if the following conditions hold: (1) The routing path uses the "Tp" API (2) The routing path only contains one single destination routing path.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00320]	

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths	
BSW Parameter		BSW Type
PduRRoutingPathGroup		EcucParamConfContainerDef
BSW Description		
<p>This container groups routing path destinations. Destinations are used instead of routing paths since a routing path can be 1:n. It is desirable to be able to enable/disable a specific bus (i.e. a destination) rather than a routing path. Of course it is possible to create groups that covers specific routing paths as well.</p> <p>Enabling and disabling of routing path groups are made using the PduR API</p>		
Template Description		
The AUTOSAR PduR will enable and disable the sending of configurable groups of IPdus during runtime according to the AUTOSAR PduR specification.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PdurIPduGroup		
Mapping Rule		Mapping Type
Create container for each existing PduRIPduGroup that is connected to the regarded Ecu		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_PduR_00308]	

BSW Module		BSW Context	
PduR		PduR/PduRRoutingPaths/PduRRoutingPathGroup	
BSW Parameter		BSW Type	
PduRIsEnabledAtInit		EcucBooleanParamDef	
BSW Description			
If set to true this routing path group will be enabled after initializing the PDU Router module (i.e. enabled in the PduR_Init function).			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00329]	

BSW Module		BSW Context	
PduR		PduR/PduRRoutingPaths/PduRRoutingPathGroup	
BSW Parameter		BSW Type	
PduRRoutingPathGroupId		EcucIntegerParamDef	
BSW Description			
Identification of the routing group.			
The identification will be used by the disable/enable API in the PDU Router module API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00309]	

BSW Module		BSW Context	
PduR		PduR/PduRRoutingPaths	
BSW Parameter		BSW Type	
PduRSrcPdu		EcucParamConfContainerDef	
BSW Description			
This container is a subcontainer of PduRRoutingPath and specifies the source of the PDU to be routed.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_PduR_00288]	

BSW Module		BSW Context	
------------	--	-------------	--

PduR	PduR/PduRRoutingPaths/PduRSrcPdu	
BSW Parameter		BSW Type
PduRSourcePduBlockSize	EcucIntegerParamDef	
BSW Description		
Minimum amount of buffer space required by receiving transport protocol layer to continue reception.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00355]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRSrcPdu	
BSW Parameter		BSW Type
PduRSourcePduHandleId	EcucIntegerParamDef	
BSW Description		
PDU identifier assigned by PDU Router.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00311]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRSrcPdu	
BSW Parameter		BSW Type
PduRSrcPduRef	EcucReferenceDef	
BSW Description		
Source PDU reference; reference to unique PDU identifier which shall be used for the requested PDU Router operation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_PduR_00318]

BSW Module	BSW Context	
PduR	PduR/PduRRoutingPaths/PduRSrcPdu	
BSW Parameter		BSW Type
PduRSrcPduUpTxConf	EcucBooleanParamDef	
BSW Description		

When enabled, the TxConfirmation will be forwarded to the upper layer. Prerequisites: Lower layer and upper layer support TxConfirmation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00351]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths
BSW Parameter	BSW Type
PduRTxBuffer	EcucParamConfContainerDef
BSW Description	
Specifies a buffer used for gatewaying via communication interfaces or transport protocols.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00336]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRTxBuffer
BSW Parameter	BSW Type
PduRPduMaxLength	EcucIntegerParamDef
BSW Description	
Length of the Tx buffer in bytes. This parameter limits the size of buffered routed PDUs.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_PduR_00324]

C.1.6 Nm Interface

BSW Module	BSW Context
Nm	Nm
BSW Parameter	BSW Type
NmChannelConfig	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of the bus channel(s). The channel parameter shall be harmonized within the whole communication stack.	

Template Description	
Set of NM nodes coordinated with use of the NM algorithm.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster	
Mapping Rule	Mapping Type
Create Container for each existing NmCluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00197]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig
BSW Parameter	BSW Type
NmActiveCoordinator	EcucBooleanParamDef
BSW Description	
This parameter indicates whether a NM channel - part of a Nm Coordination cluster - will be coordinated actively (NmActiveCoordinator = TRUE) or passively (NmActiveCoordinator = FALSE).	
Template Description	
This attribute indicates the role the NM Coordinator will have on this channel.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmCoordinatorRole	
Mapping Rule	Mapping Type
If nmCoordinatorRole is set to Active then NmActiveCoordinator shall be present and set to true. If nmCoordinatorRole is set to Passive then NmActiveCoordinator shall be present and set to false.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00236]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig
BSW Parameter	BSW Type
NmBusType	EcucChoiceContainerDef
BSW Description	
Template Description	
CanNmCluster: Can specific NmCluster attributes FlexrayNmCluster: FlexRay specific NM cluster attributes. UdpNmCluster: Udp specific NmCluster attributes	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster, SystemTemplate::NetworkManagement::FlexrayNmCluster, SystemTemplate::NetworkManagement::UdpNmCluster	
Mapping Rule	Mapping Type
Bus Type can be derived from the BusNm Configuration in the System Description.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00218]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType	
BSW Parameter	BSW Type	
NmGenericBusNmConfig	EcucParamConfContainerDef	
BSW Description		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00225]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType/NmGenericBusNmConfig	
BSW Parameter	BSW Type	
NmGenericBusNmPrefix	EcucStringParamDef	
BSW Description		
The prefix which identifies the generic <BusNm>. This will be used to determine the API name to be called by Nm for the provided interfaces of the <BusNm>. This string will used for the module prefix before the "_" character in the API call name.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00219]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType/NmGenericBusNmConfig	
BSW Parameter	BSW Type	
NmGenericBusNmShutdownTime	EcucFloatParamDef	
BSW Description		
This parameter shall be used to calculate shutdown delay time.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00239]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType	

BSW Parameter		BSW Type	
NmStandardBusNmConfig		EcucParamConfContainerDef	
BSW Description			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Nm_00226]	

BSW Module		BSW Context	
Nm		Nm/NmChannelConfig/NmBusType/NmStandardBusNmConfig	
BSW Parameter		BSW Type	
NmStandardBusType		EcucEnumerationParamDef	
BSW Description			
Identifies the bus type of the channel for standard AUTOSAR <BusNm>s and is used to determine which set of API calls to be called by Nm for the <BusNm>s. Note: The Ethernet bus' NM is UdpNm !			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Nm_00220]	

BSW Module		BSW Context	
Nm		Nm/NmChannelConfig	
BSW Parameter		BSW Type	
NmChannelSleepMaster		EcucBooleanParamDef	
BSW Description			
This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision. If this parameter is set to TRUE, the Nm shall assume that the channel is always ready to go to sleep and that no calls to Nm_RemoteSleepIndication or Nm_RemoteSleepCancellation will be made from the <BusNm> representing this channel. If this parameter is set to FALSE, the Nm shall not assume that the network is ready to sleep until a call has been made to Nm_RemoteSleepCancellation.			
Template Description			
This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision.			
M2 Parameter			
SystemTemplate::NetworkManagement::NmCluster.nmChannelSleepMaster			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	

valid	[ECUC_Nm_00227]
-------	-----------------

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter	BSW Type	
NmComMChannelRef	EcucReferenceDef	
BSW Description	Reference to the corresponding ComM Channel.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00217]	

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter	BSW Type	
NmComUserDataSupport	EcucBooleanParamDef	
BSW Description	This parameter indicates whether on a NM channel user data is accessed via Com signals or by SetUserData API.	
Template Description	Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping, SystemTemplate::NetworkManagement::NmCluster.nmPncParticipation	
Mapping Rule	Mapping Type	
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping (and is consequently handled via the PduR and Com) then NmComUserDataSupport shall be set to true. If there exists a NmCluster which has a NmNode which refers to a NmEcu and that NmEcu in turn references the EcuInstance for which this Ecu Configuration is derived and the NmCluster.nmPncParticipation has the value "true" or is not defined then NmComUserDataSupport shall be set to true.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00241]	

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter	BSW Type	
NmCoordClusterIndex	EcucIntegerParamDef	
BSW Description	If this parameter is undefined for a channel, the corresponding bus does not belong to an NM coordination cluster.	
Template Description	NmCoordinationCluster identification number.	
M2 Parameter	SystemTemplate::NetworkManagement::NmNode.nmCoordCluster	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00221]	

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		BSW Type
NmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
This parameter indicates whether a NM channel is active, e.g. can request communication and keep the bus awake, or passive, e.g. can just be woken up and kept awake by other ECUs.		
Template Description		
Enables support of the Passive Mode. The passive mode is configurable per channel.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmPassiveModeEnabled		
Mapping Rule		Mapping Type
1:1 mapping.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00242]	

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		BSW Type
NmStateReportEnabled		EcucBooleanParamDef
BSW Description		
Specifies if the NMS shall be set for the corresponding network. false: No NMS shall be set true: The NMS shall be set		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00231]	

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		BSW Type
NmStateReportSignalRef		EcucReferenceDef
BSW Description		
Reference to the signal for setting the NMS by calling Com_SendSignal for the respective channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00232]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter	BSW Type	
NmSynchronizingNetwork	EcucBooleanParamDef	
BSW Description		
If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.		
Template Description		
If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmSynchronizingNetwork		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00223]	

BSW Module	BSW Context	
Nm	Nm	
BSW Parameter	BSW Type	
NmGlobalConfig	EcucParamConfContainerDef	
BSW Description		
This container contains all global configuration parameters of the Nm Interface.		
Template Description		
A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00196]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter	BSW Type	
NmEcucPartitionRef	EcucReferenceDef	
BSW Description		
Reference to EcucPartition, where Nm module is assigned to.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Nm_00245]
-------	-----------------

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter	BSW Type	
NmGlobalConstants	EcucParamConfContainerDef	
BSW Description		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00198]

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalConstants	
BSW Parameter	BSW Type	
NmNumberOfChannels	EcucIntegerParamDef	
BSW Description		
Number of NM channels allowed within one ECU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00201]

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter	BSW Type	
NmGlobalFeatures	EcucParamConfContainerDef	
BSW Description		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Nm_00200]

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	

BSW Parameter		BSW Type	
NmBusSynchronizationEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling bus synchronization support of the <BusNm>s. This feature is required for NM Coordinator nodes only.			
Template Description			
Enables bus synchronization support.			
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Nm_00208]	

BSW Module		BSW Context	
Nm		Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		BSW Type	
NmCarWakeUpCallout		EcucFunctionNameDef	
BSW Description			
Name of the callout function to be called if Nm_CarWakeUpIndication() is called. If this parameter is not configured, the Nm will call BswM_Nm_CarWakeUpIndication.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Nm_00234]	

BSW Module		BSW Context	
Nm		Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		BSW Type	
NmCarWakeUpRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables CWU detection. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Nm_00235]	

BSW Module		BSW Context	
Nm		Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		BSW Type	

NmComControlEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00210]

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter	BSW Type	
NmCoordinatorSupportEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling NM Coordinator support.		
Template Description		
A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator		
Mapping Rule	Mapping Type	
If NmCoordinators are defined set this parameter to true.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00206]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter	BSW Type	
NmCoordinatorSyncSupport	EcucBooleanParamDef	
BSW Description		
Enables/disables the coordinator synchronisation support.		
Template Description		
Switch for enabling NmCoordinatorSync (coordination of nested busses) support.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator.nmCoordSyncSupport		
Mapping Rule	Mapping Type	
If NmCoordinator is present then the value of NmCoordinatorSyncSupport shall be set to the value of nmCoordSyncSupport.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00240]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter	BSW Type	
NmGlobalCoordinatorTime	EcucFloatParamDef	
BSW Description		

This parameter defines the maximum shutdown time of a connected and coordinated NM-Cluster. Note: This includes nested connections.	
Template Description	
This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCoordinator.nmGlobalCoordinatorTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00237]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	BSW Type
NmPduRxIndicationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the PDU Rx Indication.	
Template Description	
Switch for enabling the PDU Rx Indication.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00214]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	BSW Type
NmRemoteSleepIndEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling Remote Sleep Indication support. This feature is required for a Gateway or Nm Coordinator functionality.	
Note that this feature should not be used if all NM channels have Passive Mode enabled.	
Template Description	
Switch for enabling remote sleep indication support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00207]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	BSW Type
NmStateChangeIndEnabled	EcucBooleanParamDef
BSW Description	

Pre-processor switch for enabling the Network Management state change notification.	
Template Description	
Enables the CAN Network Management state change notification.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Nm_00215]

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		BSW Type
NmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling User Data support.		
Template Description		
Switch for enabling user data support.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00211]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter		BSW Type
NmGlobalProperties		EcucParamConfContainerDef
BSW Description		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00199]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalProperties	
BSW Parameter		BSW Type
NmCycletimeMainFunction		EcucFloatParamDef
BSW Description		
The period between successive calls to the Main Function of the NM Interface in seconds.		
Template Description		
The period between successive calls to the Main Function of the NM Interface in seconds.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmCycletimeMainFunction		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00205]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalProperties	
BSW Parameter		BSW Type
NmDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00203]	

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalProperties	
BSW Parameter		BSW Type
NmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling Version Info API support.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Nm_00204]	

C.1.7 EcuC

BSW Module	BSW Context	
EcuC	EcuC	
BSW Parameter		BSW Type
EcucConfigSet		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the global PduCollection.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00061]

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet	
BSW Parameter	BSW Type	
EcucPduCollection	EcucParamConfContainerDef	
BSW Description		
Collection of all Pdu objects flowing through the Com-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00002]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection	
BSW Parameter	BSW Type	
MetaDataType	EcucParamConfContainerDef	
BSW Description		
Meta data serves to transport information through the AUTOSAR layers. It is transported by the PduInfoType structure via a separate pointer to a byte array alongside the length of and a pointer to the payload of the PDU. This container defines the content of the meta data.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00073]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType	
BSW Parameter	BSW Type	
MetaDataTypeItem	EcucParamConfContainerDef	
BSW Description		
The content of meta data in a Pdu consists of an ordered list of meta data items. This container represents a meta data item that is contained in meta data of a Pdu.		
Template Description		
This meta-class represents a single meta-data item.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataTypeItem		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	

valid	[ECUC_EcuC_00074]
-------	-------------------

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem	
BSW Parameter	BSW Type	
MetaDataItemLength	EcucIntegerParamDef	
BSW Description		
This parameter defines the length of a meta data item in bytes.		
Template Description		
This attribute determines the length of the MetaDataItem at run-time.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.length		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00075]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem	
BSW Parameter	BSW Type	
MetaDataType	EcucEnumerationParamDef	
BSW Description		
This parameter defines the type of a meta data item.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataType		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00076]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItemMetaDataType	
BSW Parameter	BSW Type	
ADDRESS_EXTENSION_8	EcucEnumerationLiteralDef	
BSW Description		
Address extension field (N_AE) of the mixed addressing modes with 11bit and 29bit CAN ID of ISO 15765-2. Size: 8 bits.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItemMetaDataType		
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItemMetaDataType == ADDRESS_EXTENSION_16	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
CAN_ID_32	EcucEnumerationLiteralDef	
BSW Description		
CAN ID according to ISO 11898-2, either 29 bits or 11 bits. Encoding according to Can_IdType. Size: 32 bits.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem		
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == CAN_ID_32	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
ETHERNET_MAC_64	EcucEnumerationLiteralDef	
BSW Description		
Ethernet MAC address. Size: 64 bits.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem		
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == ETHERNET_MAC_64	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
LIN_NAD_8	EcucEnumerationLiteralDef	
BSW Description		
LIN node address as used in the LIN transport protocol. Size: 8 bits.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem		
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == LIN_NAD_8	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
PRIORITY_8	EcucEnumerationLiteralDef	
BSW Description	Priority field of SAE J1939 IDs, or Ethernet QoS parameter. Size: 8 bits.	
Template Description	This aggregation contributes the specification of the concrete meta-data item type.	
M2 Parameter	SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem	
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == PRIORITY_8	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
SOCKET_CONNECTION_ID_16	EcucEnumerationLiteralDef	
BSW Description	SoAd socket connection ID. Size: 16 bits.	
Template Description	This aggregation contributes the specification of the concrete meta-data item type.	
M2 Parameter	SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem	
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == SOCKET_CONNECTION_ID_16	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
SOURCE_ADDRESS_16	EcucEnumerationLiteralDef	
BSW Description	Source address of CanTp, FrTp, or DoIP transport protocol messages, or of SAE J1939 messages. Size: 16 bits.	
Template Description	This aggregation contributes the specification of the concrete meta-data item type.	
M2 Parameter	SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem	
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == SOURCE_ADDRESS_16	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem/MetaDataItem	
BSW Parameter	BSW Type	
TARGET_ADDRESS_16	EcucEnumerationLiteralDef	
BSW Description		
Target address of CanTp, FrTp, or DoIP transport protocol messages, or destination address of SAE J1939 messages. Size: 16 bits.		
Template Description		
This aggregation contributes the specification of the concrete meta-data item type.		
M2 Parameter		
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem		
Mapping Rule	Mapping Type	
SWComponentTemplate::PortInterface::MetaDataItem.metaDataItem == TARGET_ADDRESS_16	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection	
BSW Parameter	BSW Type	
Pdu	EcucParamConfContainerDef	
BSW Description		
One Pdu flowing through the COM-Stack. This Pdu is used by all Com-Stack modules to agree on referencing the same Pdu.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00001]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter	BSW Type	
DynamicLength	EcucBooleanParamDef	
BSW Description		
This parameter defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].		
Template Description		
This attribute defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.hasDynamicLength		
Mapping Rule	Mapping Type	
Attribute can be derived from Pdu.hasDynamicLength attribute that is only relevant for UserDefinedPdus, UserDefinedIPdus, J1939DcmIPdus.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuC_00078]	

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		BSW Type
EcucPduDedicatedPartition		EcucParamConfContainerDef
BSW Description		
Module specific container for Pdu to partition assignment.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00079]

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu/EcucPduDedicatedPartition	
BSW Parameter		BSW Type
EcucPduDedicatedPartitionBswModuleRef		EcucForeignReferenceDef
BSW Description		
Reference to BSW module, for which the according dedicated Pdu assignment is valid.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00080]

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu/EcucPduDedicatedPartition	
BSW Parameter		BSW Type
EcucPduDedicatedPartitionRef		EcucReferenceDef
BSW Description		
Module specific reference to EcucPartition, where the according Pdu is assigned to. The dedicated partition reference shall overrule the default partition reference for the respective module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00081]

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		BSW Type
EcucPduDefaultPartitionRef		EcucReferenceDef

BSW Description	
Reference to EcuCPartition, where the according Pdu is assigned to.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
local	
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00082]

BSW Module	BSW Context
EcuC	EcuC/EcuCConfigSet/EcuCPartition/Pdu
BSW Parameter	BSW Type
J1939Requestable	EcucBooleanParamDef
BSW Description	
Pdu can be triggered by the J1939 request message.	
Template Description	
CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message.	
J1939TpPg.requestable: Parameter Group can be triggered by the J1939 request message.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.j1939requestable, SystemTemplate::TransportProtocols::J1939TpPg.requestable	
Mapping Rule	Mapping Type
CanFrameTriggering.j1939requestable: CanFrameTriggering references a Frame where the aggregated PduToFrameMapping references the given Pdu. J1939TpPg.requestable: J1939TpPg references the given Pdu in the role sdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00072]

BSW Module	BSW Context
EcuC	EcuC/EcuCConfigSet/EcuCPartition/Pdu
BSW Parameter	BSW Type
MetaData TypeRef	EcucReferenceDef
BSW Description	
Reference to meta data that is transported in the Pdu through the AUTOSAR layers.	
Template Description	

<p>VariableDataPrototype: A VariableDataPrototype is used to contain values in an ECU application. This means that most likely a VariableDataPrototype allocates "static" memory on the ECU. In some cases optimization strategies might lead to a situation where the memory allocation can be avoided.</p> <p>In particular, the value of a VariableDataPrototype is likely to change as the ECU on which it is used executes.</p> <p>SenderReceiverToSignalMapping: Mapping of a sender receiver communication data element to a signal.</p> <p>SystemSignal: The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances.</p> <p>ISignal: 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.</p> <p>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.</p> <p>ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping).</p> <p>In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup.</p> <p>ISignalIPdu: 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.</p> <p>A maximum of one dynamic length signal per IPdu is allowed.</p>	
M2 Parameter	
SWComponentTemplate::Datatype::DataPrototypes::VariableDataPrototype SystemTemplate::DataMapping::SenderReceiverToSignalMapping SystemTemplate::Fibex::FibexCore::CoreCommunication::SystemSignal SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu	
Mapping Rule	Mapping Type
A MetaDataTypeRef shall be derived for a given Pdu if a MetaDataItemSet exists that refers to a VariablePrototype that is also referenced from a SenderReceiverToSignalMapping that in turn references a SystemSignal that is referenced by a ISignal that is mapped to an ISignalIPdu that is derived to the mentioned Pdu in EcuC.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00077]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu

BSW Parameter		BSW Type
PduLength		EcucIntegerParamDef
BSW Description		
Length of the Pdu in bytes. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits.		
Template Description		
<p>Pdu.length: Pdu length in bytes. In case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits.</p> <p>The Pdu length of zero bytes is allowed.</p> <p>IPduMapping.pduMaxLength: Define the maximum length in bytes of a Pdu which should be used as length for the corresponding sourceIPdu and targetIPdu. If IPduMapping.pduMaxLength is defined, the length for the corresponding sourceIPdu and targetIPdu shall be set to IPduMapping.pduMaxLength.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.length,		SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping.pduMaxLength
Mapping Rule		Mapping Type
1:1 mapping of Pdu.length in case that IPduMapping is not used. In case IPduMapping is used: 1:1 (sourceIPdu:targetIPdu) routing: When the SysTPduToPduTriggeringRef PduTriggering is referenced by an IPduMapping in the role sourceIPdu and targetIPdu, respectively, and that IPduMapping has a pduMaxLength defined then IPduMapping.pduMaxLength shall be used as PduLength for the derived PduRSrcPdu and PduRDestPdu, respectively. Otherwise use Pdu.length. 1:N (sourceIPdu:targetIPdu) routing: If 1:N (sourceIPdu:targetIPdu) routing is modeled, then the maximum length of the available IPduMapping.pduMaxLength and Pdu.length shall be derived for the PduRSrcPdu. The derivation of the length of each PduRDestPdu shall follow the rule for 1:1 routing. N:1 (sourceIPdu:targetIPdu): If N:1 (sourceIPdu:targetIPdu) routing is modeled, then the maximum length of the available IPduMapping.pduMaxLength and Pdu.length shall be derived for the PduRDestPdu. The derivation of the length of each PduRSrcPdu shall follow the rule for 1:1 routing.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00003]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu
BSW Parameter	
SysTPduToFrameTriggeringRef	EcucForeignReferenceDef
BSW Description	

Reference to the FrameTriggering from the SystemTemplate which this Pdu belongs to.	
SysTPduToFrameTriggeringRef shall be used for UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToPduTriggeringRef exists.	
Template Description	
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent.	
For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::FrameTriggering	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00052]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu
BSW Parameter	BSW Type
SysTPduToPduTriggeringRef	EcucForeignReferenceDef
BSW Description	
Reference to the PduTriggering from the SystemTemplate which this Pdu represents.	
SysTPduToPduTriggeringRef shall be used for all Pdus except UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. For these Pdus, SysTPduToFrameTriggeringRef shall be used.	
Template Description	
The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu.	
Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface.	
If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00054]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection
BSW Parameter	BSW Type
PdulTypeEnum	EcucEnumerationParamDef
BSW Description	

The PdulType is used within the entire AUTOSAR Com Stack except for bus drivers. The size of this global type depends on the maximum number of PDUs used within one software module. If no software module deals with more PDUs than 256, this type can be set to uint8. If at least one software module handles more than 256 PDUs, this type shall be set to uint16. See AUTOSAR_SWS_CommunicationStackTypes for more details.

Template Description
M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00041]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection

BSW Parameter	BSW Type
PduLengthTypeEnum	EcucEnumerationParamDef

BSW Description

The PduLengthType is used within the entire AUTOSAR Com Stack except for bus drivers. The size of this global type depends on the maximum length of PDUs to be sent by an ECU. If no segmentation is used the length depends on the maximum payload size of a frame of the underlying communication system (for FlexRay maximum size is 255 bytes, therefore uint8). If segmentation is used it depends on the maximum length of a segmented N-SDU (in general uint16 is used). See AUTOSAR_SWS_CommunicationStackTypes for more details.

Template Description
M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00042]

BSW Module	BSW Context
EcuC	EcuC

BSW Parameter	BSW Type
EcucHardware	EcucParamConfContainerDef

BSW Description

Hardware definition of this Ecu.

Template Description
M2 Parameter

Mapping Rule	Mapping Type
--------------	--------------

Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00056]

BSW Module	BSW Context
EcuC	EcuC/EcucHardware

BSW Parameter		BSW Type
EcucCoreDefinition		EcucParamConfContainerDef
BSW Description		
Definition of one Core on this Ecu.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00057]

BSW Module	BSW Context	
EcuC	EcuC/EcucHardware/EcucCoreDefinition	
BSW Parameter		BSW Type
EcucCoreHwRef		EcucForeignReferenceDef
BSW Description		
Optional reference to the HwElement of HwCategory ProcessingUnit that represents this Core in the ECU Resource Template.		
Template Description		
This represents the ability to describe Hardware Elements on an instance level. The particular types of hardware are distinguished by the category. This category determines the applicable attributes. The possible categories and attributes are defined in HwCategory.		
M2 Parameter		
EcuResourceTemplate::HwElement		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00059]

BSW Module	BSW Context	
EcuC	EcuC/EcucHardware/EcucCoreDefinition	
BSW Parameter		BSW Type
EcucCoreId		EcucIntegerParamDef
BSW Description		
ID of the core.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00058]

BSW Module	BSW Context	
EcuC	EcuC	
BSW Parameter		BSW Type
EcucPartitionCollection		EcucParamConfContainerDef

BSW Description	
Collection of Partitions defined for this ECU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00007]

BSW Module	BSW Context
EcuC	EcuC/EcucPartitionCollection
BSW Parameter	BSW Type
EcucPartition	EcucParamConfContainerDef
BSW Description	
Definition of one Partition on this ECU. One Partition will be implemented using one Os-Application.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00005]

BSW Module	BSW Context
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	BSW Type
EcucDefaultBswPartition	EcucBooleanParamDef
BSW Description	
Denotes the default BSW partition. This partition will host all BSW Modules, which are not explicitly mapped to a different partition. For partitions other than the default BSW partition this parameter can be omitted.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00037]

BSW Module	BSW Context
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	BSW Type
EcucPartitionBswModuleDistinguishedPartition	EcucForeignReferenceDef
BSW Description	

This maps the abstract partition of the Bsw Module to a concrete Partition existing in the ECU.	
Template Description	
Each instance of this meta-class represents an abstract partition in which context the code of the enclosing BswModuleBehavior can be executed.	
The intended use case is to distinguish between several partitions in order to implement different behavior per partition, for example to behave either as a master or satellite in a multicore ECU with shared BSW code.	
M2 Parameter	
BswModuleTemplate::BswBehavior::BswDistinguishedPartition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00068]

BSW Module	BSW Context
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	BSW Type
EcucPartitionSoftwareComponentInstanceRef	EcucInstanceReferenceDef
BSW Description	
References the SW Component instances from the Ecu Extract that shall be executed in this partition.	
Template Description	
Role of a software component within a composition.	
M2 Parameter	
SWComponentTemplate::Composition::SwComponentPrototype	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00036]

BSW Module	BSW Context
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	BSW Type
PartitionCanBeRestarted	EcucBooleanParamDef
BSW Description	
Specifies the requirement whether the Partition can be restarted. If set to true all software executing in this partition shall be capable of handling a restart.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00006]

BSW Module	BSW Context
EcuC	EcuC
BSW Parameter	BSW Type

EcucPostBuildVariants	EcucParamConfContainerDef
BSW Description	
Collection of toplevel PostBuildSelectable variants. The PredefinedVariants linked inside this container will determine how many PostBuildSelectableVariants exist. If this container exist the name pattern for initialization of BSW modules will be <Mip>_Config_<PredefinedVariant.shortName>. If this container does not exist the name pattern for initialization of BSW modlues will be <Mip>_Config.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00070]

BSW Module	BSW Context
EcuC	EcuC/EcucPostBuildVariants
BSW Parameter	BSW Type
EcucPostBuildVariantRef	EcucForeignReferenceDef
BSW Description	
Reference to a PredefinedVariant that defines one toplevel postBuild configuration set (covering all post-build capable BSW modules). PredefinedVariants that are referenced here shall contain only PostBuildVariantCriterionValueSets.	
Template Description	
This specifies one predefined variant. It is characterized by the union of all system constant values and post-build variant criterion values aggregated within all referenced system constant value sets and post build variant criterion value sets plus the value sets of the included variants.	
M2 Parameter	
GenericStructure::VariantHandling::PredefinedVariant	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00071]

BSW Module	BSW Context
EcuC	EcuC
BSW Parameter	BSW Type
EcucUnitGroupAssignment	EcucParamConfContainerDef
BSW Description	
Collection of UnitGroup references to support the generation of ASAM MCD file.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00063]

BSW Module	BSW Context
-------------------	--------------------

EcuC	EcuC/EcucUnitGroupAssignment	
BSW Parameter		BSW Type
EcucUnitGroupRef		EcucForeignReferenceDef
BSW Description		
Optional reference to the UnitGroup to support the generation of ASAM MCD file. These UnitGroups are selecting a set of units for a specific country.		
Template Description		
This meta-class represents the ability to specify a logical grouping of units. The category denotes the unit system that the referenced units are associated to.		
In this way, e.g. country-specific unit systems (CATEGORY="COUNTRY") can be defined as well as specific unit systems for certain application domains.		
In the same way a group of equivalent units, can be defined which are used in different countries, by setting CATEGORY="EQUIV_UNITS". KmPerHour and MilesPerHour could such be combined to one group named "vehicle_speed". The unit MeterPerSec would not belong to this group because it is normally not used for vehicle speed. But all of the mentioned units could be combined to one group named "speed".		
Note that the UnitGroup does not ensure the physical compliance of the units. This is maintained by the physical dimension.		
M2 Parameter		
AsamHdo::Units::UnitGroup		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00062]

BSW Module	BSW Context	
EcuC	EcuC	
BSW Parameter		BSW Type
EcucVariationResolver		EcucParamConfContainerDef
BSW Description		
Collection of PredefinedVariant elements containing definition of values for SwSystemconst which shall be applied when resolving the variability during ECU Configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuC_00009]

BSW Module	BSW Context	
EcuC	EcuC/EcucVariationResolver	
BSW Parameter		BSW Type
PredefinedVariantRef		EcucForeignReferenceDef
BSW Description		
Template Description		

This specifies one predefined variant. It is characterized by the union of all system constant values and post-build variant criterion values aggregated within all referenced system constant value sets and post build variant criterion value sets plus the value sets of the included variants.	
M2 Parameter	
GenericStructure::VariantHandling::PredefinedVariant	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuC_00010]

C.1.8 ComM

BSW Module	BSW Context
ComM	ComM
BSW Parameter	BSW Type
ComMConfigSet	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR ComM module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00879]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet
BSW Parameter	BSW Type
ComMChannel	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of the bus channel(s). The channel parameters shall be harmonized within the whole communication stack.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.commConnector	
Mapping Rule	Mapping Type
* Can, Lin, Fr: For each CommunicationCluster the EcuInstance is connected to, one ComMChannel container is created. * For Ethernet: For each EthernetPhysicalChannel the EcuInstance is connected to, one ComMChannel container is created.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00565]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type

ComMBusType	EcucEnumerationParamDef
BSW Description	
Identifies the bus type of the channel.	
Template Description	
The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.	
A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both.	
A CommunicationCluster aggregates one or more physical channels.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster	
Mapping Rule	Mapping Type
Depends of the used CommunicationCluster subclass: abstractCanCluster --> COMM_BUS_TYPE_CAN FlexRayCluster --> COMM_BUS_TYPE_FR EthernetCluster --> COMM_BUS_TYPE_ETH LinCluster --> COMM_BUS_TYPE_LIN UserDefinedCluster --> COMM_BUS_TYPE_CDD	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00567]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMCDDBusPrefix	EcucStringParamDef
BSW Description	
Prefix to be used for API calls to CDD.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00888]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMChannelId	EcucIntegerParamDef
BSW Description	
Channel identification number of the corresponding channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_ComM_00635]
-------	-------------------

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter	BSW Type	
ComMChannelPartitionRef	EcucReferenceDef	
BSW Description	Reference to EcucPartition, where the according ComMChannel is assigned to.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00894]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter	BSW Type	
ComMFullCommRequestNotificationEnabled	EcucBooleanParamDef	
BSW Description	Defines if the optional SenderReceiver Port of Interface ComM_CurrentChannelRequest will be provided for this channel. True means enabled. False means disabled	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00787]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter	BSW Type	
ComMMainFunctionPeriod	EcucFloatParamDef	
BSW Description	Specifies the period in seconds that the MainFunction has to be triggered with. Comment: ComM scheduling shall be at least as fast as the communication stack and a schedule longer than 100ms makes no sense for communication.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_ComM_00556]
-------	-------------------

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMManageReference	EcucReferenceDef
BSW Description	
Represent the reference between a ComMChannel with role managing channel and a ComMChannel with role managed channel.	
Template Description	
Reference between a channel with role managing channel and a channel with role managed channel.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.managedPhysicalChannel	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00893]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMNetworkManagement	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the networkmanagement.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00607]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement
BSW Parameter	BSW Type
ComMNmLightTimeout	EcucFloatParamDef
BSW Description	
Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left. The range shall be greater than 0.0 and less or equal to 255.0.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00606]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement	
BSW Parameter	BSW Type	
ComMNmVariant	EcucEnumerationParamDef	
BSW Description	<p>Defines the functionality of the networkmanagement.</p> <p>Shall be harmonized with NM configuration.</p>	
Template Description		
M2 Parameter	SystemTemplate::NetworkManagement::NmNode.controller	
Mapping Rule	Mapping Type	
<p>If the CommunicationController is not referenced by NmNode the ComM NmVariant of the corresponding ComMChannel shall be set to NONE if not explicitly set to LIGHT. If the CommunicationController is referenced by a Nm Node and NmEcu.nmPassiveModeEnabled attribute is present and is set to true, the ComMNm Variant shall be set to PASSIVE. If the Communication Controller is referenced by NmNode and NmEcu.nmPassiveModeEnabled attribute is not present or is set to false the ComMNmVariant shall be set to FULL.</p> <p>Set to LIN_SLAVE if the CommunicationController is defined as a Lin Slave. In the System Template the LinMaster/LinSlave is connected to the LinChannel via a CommunicationConnector.</p>	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00568]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement	
BSW Parameter	BSW Type	
ComMPncNmRequest	EcucBooleanParamDef	
BSW Description	<p>If this parameter equals true then every time a FULL Communication is requested due to a change in the PNC state machine to COMM_PNC_REQUESTED Nm shall be called using the API Nm_NetworkRequest.</p>	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00886]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter	BSW Type	
ComMNoCom	EcucBooleanParamDef	
BSW Description		

Not allowed to change state of ComM channel to COMM_SILENT_COMMUNICATION or COMM_FULL_COMMUNICATION.	
true: Enabled - Not allowed to switch to Communication Modes above. false: Disabled - Allowed to switch Communication Modes above.	
Shall be possible to change parameter during runtime with ComM API's. ECU/All channels: ComM_LimitECUToNoComMode(). Separate channels: ComM_LimitChannelToNoComMode().	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00571]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMNoWakeUpInhibitionNvmStorage	EcucBooleanParamDef
BSW Description	
If this parameter is set to "true", the NoWakeUp inhibition state of the channel shall be stored (in some implementation specific way) in the block pointed to by ComMGlobalNvmBlockDescriptor.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00789]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMNoWakeUp	EcucBooleanParamDef
BSW Description	
Defines if an ECU is not allowed to wake-up the channel. true: Enabled (not allowed to wake-up) false: Disabled	
This is the default/init value of a runtime variable that can be changed during runtime using ComM_PreventWakeUp().	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_ComM_00569]
-------	-------------------

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMPncGatewayType	EcucEnumerationParamDef
BSW Description	
Identifies the Partial Network Gateway behaviour of a ComMChannel.	
Template Description	
Defines if this EculInstance shall implement the PncGateway functionality on this Communication-Connector and its respective PhysicalChannel. Several EculInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGateway-Type "active".	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.pncGatewayType	
Mapping Rule	Mapping Type
1:1 mapping none or not defined --> do not create ECUC Parameter	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00842]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	BSW Type
ComMUserPerChannel	EcucParamConfContainerDef
BSW Description	
This container contains a list of identifiers that are needed to refer to a user in the system which is linked to a channel.	
Template Description	
ComManagementMapping.comManagementGroup: IPduGroup participating in a Mode Management PortGroup.	
ComManagementMapping.physicalChannel: This reference maps the Mode Management PortGroup partial network to communication channels.	
M2 Parameter	
SystemTemplate::ComManagementMapping.comManagementGroup,SystemTemplate::ComManagementMapping.physicalChannel	
Mapping Rule	Mapping Type

<p>The ComMUser that need to be referenced shall be derived from a superset of PhysicalChannels that are reachable by ComManagementMapping.comManagementGroup and ComManagementMapping.physicalChannel.</p> <p>From the comManagementGroup reference all ISignallPduGroups can be retrieved to which the ComManagementMapping refers to. From the ISignallPduGroup all ISignallPdus shall be collected that are contained in the ISignallPduGroup or one of the sub ISignallPduGroups.</p> <p>The search for all PduTriggerings associated with these ISignallPdus provides a set of PhysicalChannels since the PduTriggerings are directly aggregated by a PhysicalChannel.</p> <p>In addition to the PhysicalChannels that are retrieved from the ComManagementMapping.comManagementGroup the directly referenced ComManagementMapping.physicalChannel shall be added.</p> <p>Further mappings may be required from an ECU integration point of view.</p>		partial
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00657]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMUserPerChannel	
BSW Parameter	BSW Type	
ComMUserChannel	EcucReferenceDef	
BSW Description		
Reference to the ComMUser that corresponds to this channel user.		
ImplementationType: COMM_UserHandleType		
Template Description		
Mode Management PortGroup to be mapped onto a communication channel. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems.		
M2 Parameter		
SystemTemplate::ComManagementMapping.comManagementPortGroup		
Mapping Rule		Mapping Type
The ComMUser reference shall be derived from the ComMgrUserNeeds which are referenced by the ComManagementMapping.		partial
Further mappings may be required from an ECU integration point of view.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00658]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet	
BSW Parameter	BSW Type	
ComMPnc	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration of the partial network cluster (PNC).		
Template Description		

<p>@RESTRICT_TO_STANDARD:CP! Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups. @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more ServiceInstances. @END_RESTRICT_TO_STANDARD!</p>	
M2 Parameter	
SystemTemplate::PncMapping::PncMapping	
Mapping Rule	Mapping Type
Create ComMPnc container for each PncMapping element.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00843]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMPnc
BSW Parameter	BSW Type
ComMChannelPerPnc	EcucReferenceDef
BSW Description	
Reference to the ComMChannel that is required for this PNC.	
ImplementationType: NetworkHandleType	
Template Description	
<p>PncMapping.pncGroup: IPduGroup participating in a Partial Network Cluster. This reference is optional in case an ecu extract has only indirect pnc access, i.e. ecu is not directly connected to a network which supports partial network.</p> <p>PncMapping.physicalChannel: This reference maps the partial network to a communication channel.</p>	
M2 Parameter	
SystemTemplate::PncMapping::PncMapping.pncGroup,SystemTemplate::PncMapping::PncMapping.physicalChannel	
Mapping Rule	Mapping Type

<p>The ComMChannels that need to be referenced shall be derived from the PhysicalChannels that are either reachable by PncMapping.pncGroup or by PncMapping.physicalChannel.</p> <p>From the pncGroup reference all ISignalPduGroups can be retrieved to which the PncMapping refers to. From the ISignalPduGroup all ISignalPdus shall be collected that are contained in the ISignalPduGroup or one of the sub ISignalPduGroups.</p> <p>The search for all PduTriggerings associated with these ISignalPdus provides a set of PhysicalChannels since the PduTriggerings are directly aggregated by a PhysicalChannel.</p> <p>In addition to the PhysicalChannels that are retrieved from the PncMapping.pncGroup the directly referenced PncMapping.physicalChannel shall be added.</p> <p>Please note that the PncMapping.physicalChannel reference was introduced in Release 4.4.0 and for backward compatibility reasons nobody is forced to configure this new reference. Therefore the old approach via the PncMapping.pncGroup shall still be respected.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00880]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		BSW Type
ComMPncComSignal		EcucParamConfContainerDef
BSW Description		
Represents the PncComSignals which are used to communicate the EIRA and ERA status of this PNC.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00881]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter		BSW Type
ComMPncComSignalChannelRef		EcucReferenceDef
BSW Description		
Reference to the ComMChannel which is used to determine whether this PncComSignal shall participate in the active or passive role (via the parameter ComMPncGatewayType of the ComMChannel).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00884]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter	BSW Type	
ComMPncComSignalDirection	EcucEnumerationParamDef	
BSW Description		
Indicates the communication direction of this PncComSignal.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00885]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter	BSW Type	
ComMPncComSignalKind	EcucEnumerationParamDef	
BSW Description		
Indicates whether this PncComSignal represents EIRA or ERA PNC information.		
This parameter ComMPncComSignalKind is optional and shall be ignored when ComMPnc-ComSignalDirection equals TX.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00883]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter	BSW Type	
ComMPncComSignalRef	EcucReferenceDef	
BSW Description		
Reference to the ComSignal which is used to transport the partial network channel request information.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_ComM_00882]
-------	-------------------

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter	BSW Type	
ComMPncEthIfSwitchPortGroupRef	EcucReferenceDef	
BSW Description	Reference to the PortGroups that correspond to this PNC. Note: This is only for documentation.	
Template Description	Reference to the partial networks this CouplingPort participates in.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.pncMapping	
Mapping Rule	Mapping Type	
The references are derived from the reference CouplingPort to PNC_Mapping.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00891]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter	BSW Type	
ComMPncId	EcucIntegerParamDef	
BSW Description	Partial network cluster identification number.	
Template Description	Identifier of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu.	
M2 Parameter	SystemTemplate::PncMapping::PncMapping.pncIdentifier	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00874]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter	BSW Type	
ComMUserPerPnc	EcucReferenceDef	
BSW Description	Reference to the ComMUsers that correspond to this PNC.	
ImplementationType: COMM_UserHandleType		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_ComM_00876]	

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet	
BSW Parameter		BSW Type
ComMPncEnabled		EcucBooleanParamDef
BSW Description		
Defines whether in this configuration set the partial networking is enabled. true: Enabled false: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00878]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet	
BSW Parameter		BSW Type
ComMUser		EcucParamConfContainerDef
BSW Description		
This container contains a list of identifiers that are needed to refer to a user in the system which is designated to request Communication modes.		
Template Description		
Specifies the abstract needs on the configuration of the Communication Manager for one "user".		
M2 Parameter		
CommonStructure::ServiceNeeds::ComMgrUserNeeds		
Mapping Rule		Mapping Type
In case the owner of the ComMgrUserNeeds is a BSW module then the ComMUser.shortName = {capitalizedMip}_{ServiceDependency.symbolicName Props.symbol}.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00653]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMUser	
BSW Parameter		BSW Type
ComMUserEcucPartitionRef		EcucReferenceDef
BSW Description		
Denotes in which "EcucPartition" the requester is executed. When the partition is stopped, the communication request shall be cancelled in the ComM to avoid a stay-awake situation of the bus due to a stopped partition.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00786]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMUser	
BSW Parameter	BSW Type	
ComMUserIdentifier	EcucIntegerParamDef	
BSW Description		
An identifier that is needed to refer to a user in the system which is designated to request Communication Modes.		
ImplementationType: ComM_UserHandleType		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00654]

BSW Module	BSW Context	
ComM	ComM	
BSW Parameter	BSW Type	
ComMGeneral	EcucParamConfContainerDef	
BSW Description		
General configuration parameters of the Communication Manager.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00554]

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter	BSW Type	
ComM0PncVectorAvoidance	EcucBooleanParamDef	
BSW Description		
This parameter avoids sending of 0-PNC-Vectors in case ComMPncGatewayEnabled is enabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00892]

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	

BSW Parameter		BSW Type
ComMDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00555]

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		BSW Type
ComMDirectUserMapping		EcucBooleanParamDef
BSW Description		
If this parameter is set to true the configuration tool shall automatically create a ComMUser per ComMPnc and a ComMUser per ComMChannel.		
The shortName of the generated ComMUsers shall follow the following naming convention: PNCUser_ComMPncId, e.g. PNCUser_13 ChannelUser_ComMChannelId, e.g. ChannelUser_25		
Restriction: ComMUser, which are created due to this configuration parameter, shall not be used by SWCs (only available for BswM).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_ComM_00840]

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		BSW Type
ComMEcuGroupClassification		EcucIntegerParamDef
BSW Description		
Defines whether a mode inhibition affects the ECU or not.		
Examples:		
000: No mode inhibition can be activated		
001: Wake up inhibition can be enabled		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00563]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMGlobalNvMBlockDescriptor	EcucReferenceDef
BSW Description	
Reference to NVRAM block containing the none volatile data. If this parameter is not configured it means that no NVRam is used at all.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00783]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMModeLimitationEnabled	EcucBooleanParamDef
BSW Description	
true if mode limitation functionality shall be enabled. true: Enabled false: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00560]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMPncGatewayEnabled	EcucBooleanParamDef
BSW Description	
Enables or disables support of Partial Network Gateway. False: Partial Networking Gateway is disabled True: Partial Networking Gateway is enabled	
Template Description	

Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.pncGatewayType	
Mapping Rule	Mapping Type
If at least one pncGatewayType attribute is defined, then ComMPncGateway Enabled shall be set to true, if at least one CommunicationConnector of the Ecu Instance has the pncGatewayType set to either active or passive. If all pncGatewayType attributes are set to none or are not defined, the value shall be set to false.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00887]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMPncPrepareSleepTimer	EcucFloatParamDef
BSW Description	
Time in seconds the PNC state machine shall wait in COMM_PNC_PREPARE_SLEEP.	
Template Description	
Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.pncPrepareSleepTimer	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00841]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMPncSupport	EcucBooleanParamDef
BSW Description	
Enables or disables support of partial networking.	
False: Partial Networking is disabled True: Partial Networking is enabled	
Template Description	
@RESTRICT_TO_STANDARD:CP! Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups. @END_RESTRICT_TO_STANDARD!	
@RESTRICT_TO_STANDARD:AP! Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more ServiceInstances. @END_RESTRICT_TO_STANDARD!	
M2 Parameter	

SystemTemplate::PncMapping::PncMapping	
Mapping Rule	Mapping Type
If at least one Pnc is configured this parameter shall be set to true. Otherwise false.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00839]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMResetAfterForcingNoComm	EcucBooleanParamDef
BSW Description	
ComM shall perform a reset after entering "No Communication" mode because of an active mode limitation to "No Communication" mode.	
true: Enabled false: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00558]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMSynchronousWakeUp	EcucBooleanParamDef
BSW Description	
Wake up of one channel shall lead to a wake up of all channels if true.	
true: Enabled false: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00695]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMTMinFullComModeDuration	EcucFloatParamDef
BSW Description	
Minimum time duration in seconds, spent in the COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_NETWORK_REQUESTED.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00557]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMVersionInfoApi	EcucBooleanParamDef
BSW Description	
Switches the possibility to read the published information with the service ComM_GetPublishedInformation().	
true: Enabled false: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00622]

BSW Module	BSW Context
ComM	ComM/ComMGeneral
BSW Parameter	BSW Type
ComMWakeupInhibitionEnabled	EcucBooleanParamDef
BSW Description	
true if wake up inhibition functionality enabled.	
true: Enabled false: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00559]

C.1.9 Xcp

BSW Module	BSW Context
Xcp	Xcp

BSW Parameter		BSW Type
XcpConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR Xcp module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00020]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig	
BSW Parameter		BSW Type
XcpCommunicationChannel		EcucParamConfContainerDef
BSW Description		
This container represents the configuration of the communication channel of XCP.		
Template Description		
This meta-class allows to describe the relationship between several PduTriggerings that are defined on the same PhysicalChannel, e.g. to create a link between Rx and Tx Pdu that are used for request/response.		
M2 Parameter		
SystemTemplate::GeneralPurposeConnection::GeneralPurposeConnection		
Mapping Rule		Mapping Type
For each GeneralPurposeConnection of category XcpChannel one XcpCommunicationChannel shall be created.		
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00183]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpCommunicationChannel	
BSW Parameter		BSW Type
XcpChannelRxPduRef		EcucReferenceDef
BSW Description		
Optional reference to the XCP Rx PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00185]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpCommunicationChannel	
BSW Parameter		BSW Type

XcpChannelTxPduRef	EcucReferenceDef
BSW Description	
Reference to the XCP Tx PDU.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00184]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpCommunicationChannel
BSW Parameter	BSW Type
XcpComMChannelRef	EcucReferenceDef
BSW Description	
Reference to the ComM channel the PDUs belong to.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00186]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig
BSW Parameter	BSW Type
XcpDaqList	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of the DAQs.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00050]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList
BSW Parameter	BSW Type
XcpDaqListNumber	EcucIntegerParamDef
BSW Description	
Index number of the DAQ list	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00051]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList
BSW Parameter	BSW Type
XcpDaqListType	EcucEnumerationParamDef
BSW Description	
This indicates whether this DAQ list represents a DAQ or a STIM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00052]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList
BSW Parameter	BSW Type
XcpDto	EcucParamConfContainerDef
BSW Description	
This container collects data transfer object specific parameters for the DAQ list.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00065]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList/XcpDto
BSW Parameter	BSW Type
XcpDto2PduMapping	EcucChoiceReferenceDef
BSW Description	
This reference specifies the mapping of the DTO to the PDUs from the lower-layer interfaces (CanIf, Frlf, SoAd and Cdd).	
A reference to a XcpRxPdu is only feasible if the the DaqListType is DAQ_STIM. A reference to a XcpTxPdu is only feasible if the DaqListType is DAQ.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00067]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpDto	
BSW Parameter		BSW Type
XcpDtoPid		EcucIntegerParamDef
BSW Description		
Packet identifier (PID) of the DTO that identifies the ODT the content of the DTO.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00066]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter		BSW Type
XcpMaxOdt		EcucIntegerParamDef
BSW Description		
MAX_ODT indicates the maximum amount of ODTs in this DAQ list (STATIC configuration)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00053]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter		BSW Type
XcpMaxOdtEntries		EcucIntegerParamDef
BSW Description		
This parameter indicates the maximum amount of entries in an ODT of this DAQ list (STATIC configuration).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00058]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter	BSW Type	
XcpOdt	EcucParamConfContainerDef	
BSW Description	This container contains ODT-specific parameter for the DAQ list.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00055]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt	
BSW Parameter	BSW Type	
XcpOdt2DtoMapping	EcucReferenceDef	
BSW Description	This reference maps the ODT to the according DTO in which it will be transmitted.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00056]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt	
BSW Parameter	BSW Type	
XcpOdtEntry	EcucParamConfContainerDef	
BSW Description	This container collects all configuration parameters that comprise an ODT entry.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00061]	

BSW Module	BSW Context

Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		BSW Type
XcpOdtEntryAddress	EcucLinkerSymbolDef	
BSW Description		
Memory address that the ODT entry is referencing to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00063]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		BSW Type
XcpOdtEntryBitOffset	EcucIntegerParamDef	
BSW Description		
Represent the bit offset in case of the element represents status bit.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00179]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		BSW Type
XcpOdtEntryLength	EcucIntegerParamDef	
BSW Description		
Length of the referenced memory area that is referenced by the ODT entry.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00064]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		BSW Type
XcpOdtEntryNumber	EcucIntegerParamDef	
BSW Description		
Index number of the ODT entry		

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00062]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt
BSW Parameter	BSW Type
XcpOdtEntryMaxSize	EcucIntegerParamDef
BSW Description	
This parameter indicates the upper limit for the size of the element described by an ODT entry. Depending on the DaqListType this ODT belongs to it describes the limit for a DAQ (MAX_ODT_ENTRY_SIZE_DAQ) or a STIM (MAX_ODT_ENTRY_SIZE_STIM).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00060]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt
BSW Parameter	BSW Type
XcpOdtNumber	EcucIntegerParamDef
BSW Description	
Index number of this ODT within the DAQ list.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00057]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig
BSW Parameter	BSW Type
XcpEventChannel	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of event channels on the XCP slave.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00150]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelConsistency		EcucEnumerationParamDef
BSW Description		
Type of consistency used by event channel		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00171]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelMaxDaqList		EcucIntegerParamDef
BSW Description		
Maximum amount of DAQ lists that are handled by this event channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00153]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelNumber		EcucIntegerParamDef
BSW Description		
Index number of the event channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00152]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelPriority		EcucIntegerParamDef
BSW Description		
Priority of the event channel		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00154]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelTimeCycle		EcucIntegerParamDef
BSW Description		
The event channel time cycle indicates which sampling period is used to process this event channel. A value of 0 means 'Not cyclic'.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00173]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelTimeUnit		EcucEnumerationParamDef
BSW Description		
This configuration parameter indicates the unit of the event channel time cycle.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00174]

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		BSW Type
XcpEventChannelTriggeredDaqListRef		EcucReferenceDef

BSW Description	
References all DAQ lists that are triggered by this event channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00151]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpEventChannel
BSW Parameter	BSW Type
XcpEventChannelType	EcucEnumerationParamDef
BSW Description	
This configuration parameter indicates what kind of DAQ list can be allocated to this event channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00172]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig
BSW Parameter	BSW Type
XcpPageSwitching	EcucParamConfContainerDef
BSW Description	
This container represents configuration of the page switching feature.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00187]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPageSwitching
BSW Parameter	BSW Type
XcpSegment	EcucParamConfContainerDef
BSW Description	
This container represents configuration of the page switching segment element.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00188]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpPageSwitching/XcpSegment	
BSW Parameter		BSW Type
XcpPage		EcucParamConfContainerDef
BSW Description		
This container represents configuration of the optional page element.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00192]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpPageSwitching/XcpSegment/XcpPage	
BSW Parameter		BSW Type
XcpPageAddress		EcucLinkerSymbolDef
BSW Description		
Memory address of the optional page (Page ID = 2 ... 255).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00193]	

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpPageSwitching/XcpSegment	
BSW Parameter		BSW Type
XcpReferencePageAddress		EcucLinkerSymbolDef
BSW Description		
Memory address of the reference page (Page ID = 0).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00189]	

BSW Module		BSW Context	
Xcp		Xcp/XcpConfig/XcpPageSwitching/XcpSegment	
BSW Parameter		BSW Type	
XcpSegmentLength		EcucIntegerParamDef	
BSW Description			
Length of the segment in bytes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00191]	

BSW Module		BSW Context	
Xcp		Xcp/XcpConfig/XcpPageSwitching/XcpSegment	
BSW Parameter		BSW Type	
XcpWorkingPageAddress		EcucLinkerSymbolDef	
BSW Description			
Memory address address of the working page (Page ID = 1).			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00190]	

BSW Module		BSW Context	
Xcp		Xcp/XcpConfig	
BSW Parameter		BSW Type	
XcpPdu		EcucChoiceContainerDef	
BSW Description			
Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.			
Template Description			
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::GeneralPurposeIPdu			
Mapping Rule			Mapping Type
Create this container if a GeneralPurposeIPdu with the category "Xcp" is defined in the Ecu Extract.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00100]	

BSW Module		BSW Context	
Xcp		Xcp/XcpConfig/XcpPdu	
BSW Parameter		BSW Type	

XcpRxPdu	EcucParamConfContainerDef
BSW Description	
This container specifies received PDUs.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort	
Mapping Rule	Mapping Type
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the XcpPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00105]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu/XcpRxPdu
BSW Parameter	BSW Type
XcpRxPduId	EcucIntegerParamDef
BSW Description	
ID of the PDU that will be received via a Xcp_<module>RxIndication.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00106]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu/XcpRxPdu
BSW Parameter	BSW Type
XcpRxPduRef	EcucReferenceDef
BSW Description	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00107]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu
BSW Parameter	BSW Type
XcpTxPdu	EcucParamConfContainerDef
BSW Description	
This container specifies transmission PDUs.	
Template Description	

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort	
Mapping Rule	Mapping Type
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the XcpPdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00101]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu/XcpTxPdu
BSW Parameter	BSW Type
XcpTxPduId	EcucIntegerParamDef
BSW Description	
The PDU identifier, which has to be used by the lower layer BSW module for TxConfirmations or TriggerTransmits.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00103]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu/XcpTxPdu
BSW Parameter	BSW Type
XcpTxPduRef	EcucReferenceDef
BSW Description	
Reference to the external PDU definition.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00104]

BSW Module	BSW Context
Xcp	Xcp
BSW Parameter	BSW Type
XcpGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the XCP.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00001]	

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		BSW Type
XcpCounterRef		EcucReferenceDef
BSW Description		
This parameter contains a reference to the counter, which is used by XCP.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00162]	

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		BSW Type
XcpDaqConfigType		EcucEnumerationParamDef
BSW Description		
Sets the DAQ_CONFIG_TYPE bit within the DAQ_PROPERTIES parameter to "static" or to "dynamic". If DAQ_STATIC is selected, the DAQ_CONFIG_TYPE bit is set to "0". If DAQ_DYNAMIC is selected, the DAQ_CONFIG_TYPE bit is set to "1".		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Xcp_00164]	

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		BSW Type
XcpDaqCount		EcucIntegerParamDef
BSW Description		
Indicates the number of DAQ lists for dynamic configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Xcp_00012]
-------	------------------

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter	BSW Type	
XcpDevErrorDetect	EcucBooleanParamDef	
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00003]

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter	BSW Type	
XcpFlashProgrammingEnabled	EcucBooleanParamDef	
BSW Description		
Enabling of XCP Flash programming functionality		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00181]

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter	BSW Type	
XcpIdentificationFieldType	EcucEnumerationParamDef	
BSW Description		
Type of Identification Field the slave will use when transferring DAQ Packets to the master. The master has to use the same Type of Identification Field when transferring STIM Packets to the slave.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Xcp_00170]

BSW Module		BSW Context	
Xcp		Xcp/XcpGeneral	
BSW Parameter		BSW Type	
XcpMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
The XCP does not require this information but the BSW scheduler, which invokes the main function, needs it in order to plan its tasks.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00014]	

BSW Module		BSW Context	
Xcp		Xcp/XcpGeneral	
BSW Parameter		BSW Type	
XcpMaxCto		EcucIntegerParamDef	
BSW Description			
MAX_CTO shows the maximum length of a CTO packet in bytes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00004]	

BSW Module		BSW Context	
Xcp		Xcp/XcpGeneral	
BSW Parameter		BSW Type	
XcpMaxDto		EcucIntegerParamDef	
BSW Description			
MAX_DTO shows the maximum length of a DTO packet in bytes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Xcp_00005]	

BSW Module		BSW Context	
Xcp		Xcp/XcpGeneral	
BSW Parameter		BSW Type	
XcpMaxEventChannel		EcucIntegerParamDef	

BSW Description	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00011]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpMinDaq	EcucIntegerParamDef
BSW Description	
Indicates the number of predefined, read only DAQ lists on the XCP slave.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00013]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpNvRamBlockIdRef	EcucReferenceDef
BSW Description	
This reference contains the link to a non-volatile memory block to be used in the feature "RESUME MODE" so this information has to be stored non volatile to be available directly after start-up of the ECU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00180]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOdtCount	EcucIntegerParamDef
BSW Description	
This parameter indicates the amount of ODTs of a DAQ list using dynamic DAQ list configuration.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00054]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOdtEntriesCount	EcucIntegerParamDef
BSW Description	
Indicates the amount of entries into an ODT using dynamic DAQ list configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00059]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOdtEntrySizeDaq	EcucIntegerParamDef
BSW Description	
Indicates the size of an element described by an ODT entry to the DaqListType for a DAQ.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00177]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOdtEntrySizeStim	EcucIntegerParamDef
BSW Description	
Indicates the size of an element described by an ODT entry to the DaqListType for a stim.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00178]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOnCanEnabled	EcucBooleanParamDef
BSW Description	
Enabling of XCPonCAN functionality	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00006]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOnCddEnabled	EcucBooleanParamDef
BSW Description	
Enabling of XCPonCdd functionality	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00009]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpOnEthernetEnabled	EcucBooleanParamDef
BSW Description	
Enabling of XCPonEthernet functionality	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00008]

BSW Module	BSW Context

Xcp	Xcp/XcpGeneral
BSW Parameter	
XcpOnFlexRayEnabled	
BSW Type	
EcucBooleanParamDef	
BSW Description	
Enabling of XCPonFlexRay functionality	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	
ECUC Parameter ID	
[ECUC_Xcp_00007]	

BSW Module	
Xcp	
BSW Context	
Xcp/XcpGeneral	
BSW Parameter	
XcpPrescalerSupported	
BSW Type	
EcucBooleanParamDef	
BSW Description	
This parameter enables and disables the support for Prescaler support. True is Enabled, False is disabled	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	
ECUC Parameter ID	
[ECUC_Xcp_00169]	

BSW Module	
Xcp	
BSW Context	
Xcp/XcpGeneral	
BSW Parameter	
XcpSuppressTxSupport	
BSW Type	
EcucBooleanParamDef	
BSW Description	
Switches the support of suppressing transmission of PDUs per communication channel on or off. TRUE: Suppressing of TxPDUs supported FALSE: Suppressing of TxPDUs not supported	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	
ECUC Parameter ID	
[ECUC_Xcp_00176]	

BSW Module	
Xcp	
BSW Context	
Xcp/XcpGeneral	
BSW Parameter	
BSW Type	

XcpTimestampTicks	EcucIntegerParamDef
BSW Description	
This parameter defines the timestamp that will increment based <code>TIMESTAMP_TICKS</code> per unit and wrap around if an overflow occurs.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00167]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpTimestampType	EcucEnumerationParamDef
BSW Description	
This parameter indicates the number of bytes used for the timestamp field. In case <code>No_TIME_STAMP</code> is selected the timestamp field is not available.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00166]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpTimestampUnit	EcucEnumerationParamDef
BSW Description	
This parameter indicates the resolution of the data acquisition clock of the slave when transferring data to master.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Xcp_00168]

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	BSW Type
XcpVersionInfoApi	EcucBooleanParamDef
BSW Description	

Enables/disables the existence of the XCP_GetVersionInfo() API service.	
TRUE: XCP_GetVersionInfo() API service exists FALSE: XCP_GetVersionInfo() API service does not exist	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xcp_00002]

C.1.10 Bus Mirroring

BSW Module	BSW Context
Mirror	Mirror
BSW Parameter	BSW Type
MirrorConfigSet	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters and sub containers of the Bus Mirroring module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00008]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet
BSW Parameter	BSW Type
MirrorDestNetwork	EcucChoiceContainerDef
BSW Description	
Destination bus to which frames are sent by the Bus Mirroring module.	
Template Description	
This element defines a bus mirroring in which the traffic from one communication bus (sourceChannel) is forwarded to another one (targetChannel).	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMapping	
Mapping Rule	Mapping Type
Create a container for each BusMirrorChannel that is composed by an instance of a concrete subclass of BusMirrorChannelMapping in role targetChannel which is available in the System Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00051]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork

BSW Parameter		BSW Type	
MirrorDestNetworkCan		EcucParamConfContainerDef	
BSW Description			
Destination bus representing a CAN network.			
Template Description			
This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel.			
M2 Parameter			
SystemTemplate::BusMirror::BusMirrorChannelMappingCan			
Mapping Rule			Mapping Type
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirroChannelMappingCan in the role targetChannel which is available in the SystemExtract.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Mirror_00052]	

BSW Module		BSW Context	
Mirror		Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan	
BSW Parameter		BSW Type	
MirrorComMNetworkHandleRef		EcucReferenceDef	
BSW Description			
Reference to the ComMChannel that represents the bus.			
Template Description			
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.			
M2 Parameter			
SystemTemplate::BusMirror::BusMirrorChannel.channel			
Mapping Rule			Mapping Type
This reference shall be derived from the: <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIpc container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Mirror_00064]	

BSW Module		BSW Context	
Mirror		Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan	
BSW Parameter		BSW Type	
MirrorDestPdu		EcucParamConfContainerDef	
BSW Description			
I-PDU used for transmission of the mirrored frames on the destination bus.			
Template Description			

Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	Mapping Type
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00055]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduId	EcucIntegerParamDef
BSW Description	
I-PDU identifier used for TxConfirmation from PduR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00057]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the I-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00056]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduUsesTriggerTransmit	EcucBooleanParamDef

BSW Description	
Switches transmission via TriggerTransmit. * true: The I-PDU is transmitted using TriggerTransmit. * false: The I-PDU is transmitted directly with the Transmit call.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00063]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	BSW Type
MirrorDestQueueSize	EcucIntegerParamDef
BSW Description	
Number of frames that can be stored in the output queue for the destination bus.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00054]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	BSW Type
MirrorNetworkId	EcucIntegerParamDef
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	Mapping Type

<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworkIpl container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingIpl.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role source Channel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	BSW Type
MirrorStatusCanId	EcucIntegerParamDef
BSW Description	
CAN ID of the CAN status frame.	
If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
Template Description	
CAN ID of the CAN status frame.	
If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMappingCan.mirrorStatusCanId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00061]

BSW Module	BSW Context
-------------------	--------------------

Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork	
BSW Parameter		BSW Type
MirrorDestNetworkCdd	EcucParamConfContainerDef	
BSW Description		
Destination bus representing a user defined network.		
Template Description		
This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a UserDefined targetChannel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannelMappingUserDefined		
Mapping Rule		Mapping Type
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirroChannelMappingUserDefined in the role targetChannel which is available in the SystemExtract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00062]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd	
BSW Parameter		BSW Type
MirrorComMNetworkHandleRef	EcucReferenceDef	
BSW Description		
Reference to the ComMChannel that represents the bus.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannel.channel		
Mapping Rule		Mapping Type
This reference shall be derived from the: <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIpc container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00064]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd	
BSW Parameter		BSW Type
MirrorDestPdu	EcucParamConfContainerDef	
BSW Description		
I-PDU used for transmission of the mirrored frames on the destination bus.		
Template Description		

Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	Mapping Type
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00055]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduId	EcucIntegerParamDef
BSW Description	
I-PDU identifier used for TxConfirmation from PduR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00057]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the I-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00056]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd/MirrorDestPdu
BSW Parameter	BSW Type
MirrorDestPduUsesTriggerTransmit	EcucBooleanParamDef

BSW Description	
Switches transmission via TriggerTransmit. * true: The I-PDU is transmitted using TriggerTransmit. * false: The I-PDU is transmitted directly with the Transmit call.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00063]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	BSW Type
MirrorDestQueueSize	EcucIntegerParamDef
BSW Description	
Number of frames that can be stored in the output queue for the destination bus.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00054]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	BSW Type
MirrorDestTransmissionDeadline	EcucFloatParamDef
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655.35ms.	
Template Description	

BusMirrorChannelMappingFlexray.transmissionDeadline:

Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.

If omitted, destination frames are only sent when full or when the time stamp overflows.

BusMirrorChannelMappingIp.transmissionDeadline:

Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.

If omitted, destination frames are only sent when full or when the time stamp overflows.

BusMirrorChannelMappingUserDefined.transmissionDeadline:

Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.

If omitted, destination frames are only sent when full or when the time stamp overflows.

M2 Parameter

SystemTemplate::BusMirror::BusMirrorChannelMappingFlexray.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingIp.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingUserDefined.transmissionDeadline

Mapping Rule

Please note that this parameter is aggregated in different containers:

- if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMappingFlexray.transmissionDeadline
- if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmissionDeadline
- if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUserDefined.transmissionDeadline

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_Mirror_00059]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	BSW Type
MirrorNetworkId	EcucIntegerParamDef
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	Mapping Type

<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworklp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappinglp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role source Channel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourcet Channel by the instance of BusMirrorChannelMapping.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork
BSW Parameter	BSW Type
MirrorDestNetworkFlexRay	EcucParamConfContainerDef
BSW Description	
Destination bus representing a FlexRay network.	
Template Description	
This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMappingFlexray	
Mapping Rule	Mapping Type
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirroChannelMappingFlexray in the role targetChannel which is available in the SystemExtract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00058]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	BSW Type
MirrorComMNetworkHandleRef	EcucReferenceDef

BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannel.channel	
Mapping Rule	Mapping Type
This reference shall be derived from the: <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIpc container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	BSW Type
MirrorDestPduFlexRay	EcucParamConfContainerDef
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus. For FlexRay, an arbitrary number of I-PDUs can be configured.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	Mapping Type
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00066]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay/MirrorDestPduFlexRay
BSW Parameter	BSW Type
MirrorDestPduId	EcucIntegerParamDef
BSW Description	
I-PDU identifier used for TxConfirmation from PduR.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00057]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay/MirrorDestPduFlexRay
BSW Parameter	BSW Type
MirrorDestPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the I-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00056]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay/MirrorDestPduFlexRay
BSW Parameter	BSW Type
MirrorDestPduUsesTriggerTransmit	EcucBooleanParamDef
BSW Description	
Switches transmission via TriggerTransmit. * true: The I-PDU is transmitted using TriggerTransmit. * false: The I-PDU is transmitted directly with the Transmit call.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00063]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	BSW Type
MirrorDestQueueSize	EcucIntegerParamDef
BSW Description	
Number of frames that can be stored in the output queue for the destination bus.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00054]	

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	BSW Type
MirrorDestTransmissionDeadline	EcucFloatParamDef
BSW Description	
<p>Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest.</p> <p>If omitted, destination frames are only sent when full or when the time stamp overflows after 655.35ms.</p>	
Template Description	
<p>BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.</p> <p>If omitted, destination frames are only sent when full or when the time stamp overflows.</p>	
<p>BusMirrorChannelMappingIp.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.</p> <p>If omitted, destination frames are only sent when full or when the time stamp overflows.</p>	
<p>BusMirrorChannelMappingUserDefined.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.</p> <p>If omitted, destination frames are only sent when full or when the time stamp overflows.</p>	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMappingFlexray.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingIp.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingUserDefined.transmissionDeadline	
Mapping Rule	Mapping Type
<p>Please note that this parameter is aggregated in different containers:</p> <ul style="list-style-type: none"> - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMappingFlexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmissionDeadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUserDefined.transmissionDeadline 	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00059]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay	
BSW Parameter	BSW Type	
MirrorNetworkId	EcucIntegerParamDef	
BSW Description	Network ID of the bus.	
Template Description	This attribute defines the networkId of the communication channel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	Mapping Type	
<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworklp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappinglp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p>	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00012]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork	
BSW Parameter	BSW Type	
MirrorDestNetworklp	EcucParamConfContainerDef	
BSW Description	Destination bus representing an IP network.	
Template Description	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorChannelMappinglp	
Mapping Rule	Mapping Type	

Create a container for each BusMirrorChannel that is composed by an instance of a BusMirroChannelMappingIp in the role targetChannel which is available in the SystemExtract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00060]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp	
BSW Parameter	BSW Type	
MirrorComMNetworkHandleRef	EcucReferenceDef	
BSW Description		
Reference to the ComMChannel that represents the bus.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannel.channel		
Mapping Rule		Mapping Type
<p>This reference shall be derived from the:</p> <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00064]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp	
BSW Parameter	BSW Type	
MirrorDestPdu	EcucParamConfContainerDef	
BSW Description		
I-PDU used for transmission of the mirrored frames on the destination bus.		
Template Description		
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannelMapping.targetPduTriggering		
Mapping Rule		Mapping Type
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00055]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp/MirrorDestPdu	
BSW Parameter	BSW Type	
MirrorDestPduId	EcucIntegerParamDef	
BSW Description	I-PDU identifier used for TxConfirmation from PduR.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00057]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp/MirrorDestPdu	
BSW Parameter	BSW Type	
MirrorDestPduRef	EcucReferenceDef	
BSW Description	Reference to the Pdu object representing the I-PDU.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00056]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp/MirrorDestPdu	
BSW Parameter	BSW Type	
MirrorDestPduUsesTriggerTransmit	EcucBooleanParamDef	
BSW Description	Switches transmission via TriggerTransmit. * true: The I-PDU is transmitted using TriggerTransmit. * false: The I-PDU is transmitted directly with the Transmit call.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00063]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp	
BSW Parameter	BSW Type	

MirrorDestQueueSize	EcucIntegerParamDef
BSW Description	
Number of frames that can be stored in the output queue for the destination bus.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
local	
Mapping Status	
ECUC Parameter ID	
valid	[ECUC_Mirror_00054]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp
BSW Parameter	BSW Type
MirrorDestTransmissionDeadline	EcucFloatParamDef
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest.	
If omitted, destination frames are only sent when full or when the time stamp overflows after 655.35ms.	
Template Description	
BusMirrorChannelMappingFlexray.transmissionDeadline:	
Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.	
If omitted, destination frames are only sent when full or when the time stamp overflows.	
BusMirrorChannelMappingIp.transmissionDeadline:	
Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.	
If omitted, destination frames are only sent when full or when the time stamp overflows.	
BusMirrorChannelMappingUserDefined.transmissionDeadline:	
Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest.	
If omitted, destination frames are only sent when full or when the time stamp overflows.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMappingFlexray.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingIp.transmissionDeadline, SystemTemplate::BusMirror::BusMirrorChannelMappingUserDefined.transmissionDeadline	
Mapping Rule	
Mapping Type	

Please note that this parameter is aggregated in different containers:		full
- if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMappingFlexray.transmissionDeadline		
- if aggregated by MirrorDestNetworklp take value from BusMirrorChannelMappinglp.transmissionDeadline		
- if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUserDefined.transmissionDeadline		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00059]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworklp	
BSW Parameter	BSW Type	
MirrorNetworkId	EcucIntegerParamDef	
BSW Description	Network ID of the bus.	
Template Description	This attribute defines the networkId of the communication channel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	Mapping Type	
<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworklp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappinglp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannelMapping.</p>	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00012]	

BSW Module		BSW Context	
Mirror		Mirror/MirrorConfigSet	
BSW Parameter		BSW Type	
MirrorInitialDestNetworkRef		EcucReferenceDef	
BSW Description			
Reference to the destination bus that is selected after initialization of the Bus Mirroring module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Mirror_00007]	

BSW Module		BSW Context	
Mirror		Mirror/MirrorConfigSet	
BSW Parameter		BSW Type	
MirrorSourceNetwork		EcucChoiceContainerDef	
BSW Description			
Source bus from which frames are received by the Bus Mirroring module.			
Template Description			
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.			
M2 Parameter			
SystemTemplate::BusMirror::BusMirrorChannel.channel			
Mapping Rule			Mapping Type
Create a container for each BusMirrorChannel that is composed by an instance of a concrete subclass of BusMirrorChannelMapping in role sourceChannel which is available in the System Extract.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Mirror_00009]	

BSW Module		BSW Context	
Mirror		Mirror/MirrorConfigSet/MirrorSourceNetwork	
BSW Parameter		BSW Type	
MirrorSourceNetworkCan		EcucParamConfContainerDef	
BSW Description			
Source bus representing a CAN network.			
Template Description			
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.			
M2 Parameter			
SystemTemplate::BusMirror::BusMirrorChannel.channel			
Mapping Rule			Mapping Type
Create a container for each CanPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Mirror_00010]	

BSW Module	BSW Context
------------	-------------

Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter		BSW Type
MirrorComMNetworkHandleRef	EcucReferenceDef	
BSW Description		
Reference to the ComMChannel that represents the bus.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannel.channel		
Mapping Rule		Mapping Type
<p>This reference shall be derived from the:</p> <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIpc container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00064]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter		BSW Type
MirrorNetworkId	EcucIntegerParamDef	
BSW Description		
Network ID of the bus.		
Template Description		
This attribute defines the networkId of the communication channel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId		
Mapping Rule		Mapping Type

<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworklp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappinglp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role source Channel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourcet Channel by the instance of BusMirrorChannelMapping.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan
BSW Parameter	BSW Type
MirrorSourceCanFilter	EcucChoiceContainerDef
BSW Description	
Pre-configured filter for CAN frames.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00014]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter
BSW Parameter	BSW Type
MirrorSourceCanFilterMask	EcucParamConfContainerDef
BSW Description	
Pre-configured mask based filter for CAN frames.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00019]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterMask
BSW Parameter	BSW Type
MirrorSourceCanFilterCanIdCode	EcucIntegerParamDef
BSW Description	
Value to match masked CAN IDs.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00020]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterMask
BSW Parameter	BSW Type
MirrorSourceCanFilterCanIdMask	EcucIntegerParamDef
BSW Description	
Mask applied to CAN IDs before comparison.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00021]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterMask
BSW Parameter	BSW Type
MirrorSourceCanFilterId	EcucIntegerParamDef
BSW Description	
Unique identifier of the pre-configured CAN filter.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00018]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter
BSW Parameter	BSW Type
MirrorSourceCanFilterRange	EcucParamConfContainerDef
BSW Description	
Pre-configured range filter for CAN frames.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00015]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterRange
BSW Parameter	BSW Type
MirrorSourceCanFilterId	EcucIntegerParamDef
BSW Description	
Unique identifier of the pre-configured CAN filter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00018]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterRange
BSW Parameter	BSW Type
MirrorSourceCanFilterLower	EcucIntegerParamDef
BSW Description	
Lowest CAN ID that is accepted by the filter.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00016]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanFilter/MirrorSourceCanFilterRange	
BSW Parameter	BSW Type	
MirrorSourceCanFilterUpper	EcucIntegerParamDef	
BSW Description		
Highest CAN ID that is accepted by the filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00017]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter	BSW Type	
MirrorSourceCanMaskBasedIdMapping	EcucParamConfContainerDef	
BSW Description		
Rule for remapping a set of CAN IDs.		
Template Description		
This element defines a rule for remapping a set of CAN IDs.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorCanIdRangeMapping		
Mapping Rule		Mapping Type
Create Container in case that BusMirrorCanIdRangeMapping is aggregated by BusMirrorChannelMappingCan in the role canIdRangeMapping.		full
Mapping Status	ECUC Parameter ID	
obsolete	[ECUC_Mirror_00025]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping	
BSW Parameter	BSW Type	
MirrorSourceCanMaskBasedIdMappingDestBaselD	EcucIntegerParamDef	
BSW Description		
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.		
Template Description		
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorCanIdRangeMapping.destinationBaselD		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00028]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping	
BSW Parameter		BSW Type
MirrorSourceCanMaskBasedIdMappingSourceCanIdCode		EcucIntegerParamDef
BSW Description		
Value to match masked original CAN IDs.		
Template Description		
Value to match masked original CAN IDs.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorCanIdRangeMapping.sourceCanIdCode		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00026]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping	
BSW Parameter		BSW Type
MirrorSourceCanMaskBasedIdMappingSourceCanIdMask		EcucIntegerParamDef
BSW Description		
Mask applied to original CAN IDs before comparison.		
Template Description		
Mask applied to original CAN IDs before comparison.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorCanIdRangeMapping.sourceCanIdMask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00027]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter		BSW Type
MirrorSourceCanSingleIdMapping		EcucParamConfContainerDef
BSW Description		
Rule for remapping a single CAN ID.		
Template Description		
This element defines a rule for remapping a single CAN ID.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorCanIdToCanIdMapping		
Mapping Rule		Mapping Type
Create container in case that the BusMirrorCanIdToCanIdMapping is aggregated by BusMirrorChannelMappingCan in the role canIdToCanIdMapping.		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00022]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanSingleIdMapping	
BSW Parameter	BSW Type	
MirrorSourceCanSingleIdMappingDestCanId	EcucIntegerParamDef	
BSW Description	Mapped CAN ID.	
Template Description	This attribute defines the CanId on the targetChannel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorCanIdToCanIdMapping.remappedCanId	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00024]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanSingleIdMapping	
BSW Parameter	BSW Type	
MirrorSourceCanSingleIdMappingSourceCanId	EcucIntegerParamDef	
BSW Description	Original CAN ID.	
Template Description	This reference points to the sourceFrame with sourceCanId on the sourceChannel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorCanIdToCanIdMapping.souceCanId	
Mapping Rule	Mapping Type	
Take the value from the identifier attribute of the referenced CanFrameTriggering.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00023]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter	BSW Type	
MirrorSourceMaxDynamicFilters	EcucIntegerParamDef	
BSW Description	Maximum number of filters that can be dynamically added using Mirror_AddXxxFilter().	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00013]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork	
BSW Parameter	BSW Type	
MirrorSourceNetworkFlexRay	EcucParamConfContainerDef	
BSW Description	Source bus representing a FlexRay network.	
Template Description	Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorChannel.channel	
Mapping Rule	Mapping Type	
Create a container for each FlexrayPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00042]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay	
BSW Parameter	BSW Type	
MirrorComMNetworkHandleRef	EcucReferenceDef	
BSW Description	Reference to the ComMChannel that represents the bus.	
Template Description	Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	SystemTemplate::BusMirror::BusMirrorChannel.channel	
Mapping Rule	Mapping Type	
This reference shall be derived from the: <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIpc container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00064]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay	
BSW Parameter	BSW Type	
MirrorNetworkId	EcucIntegerParamDef	
BSW Description	Network ID of the bus.	
Template Description	This attribute defines the networkId of the communication channel.	
M2 Parameter		

SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	Mapping Type
<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworkItp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingItp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role target Channel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role source Channel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay
BSW Parameter	BSW Type
MirrorSourceFlexRayFilter	EcucParamConfContainerDef
BSW Description	Pre-configured filter for FlexRay frames.
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00043]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterChannelAssignment	EcucEnumerationParamDef

BSW Description	
FlexRay channels accepted by the filter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00049]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterCycleRepetition	EcucIntegerParamDef
BSW Description	
Cycle repetition of accepted cycles.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00048]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterId	EcucIntegerParamDef
BSW Description	
Unique identifier of the pre-configured FlexRay filter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00050]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterLowerBaseCycle	EcucIntegerParamDef
BSW Description	
Lowest base cycle number that is accepted by the filter.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00046]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterLowerSlot	EcucIntegerParamDef
BSW Description	
Lowest slot ID that is accepted by the filter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00044]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterUpperBaseCycle	EcucIntegerParamDef
BSW Description	
Highest base cycle number that is accepted by the filter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00047]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay/ MirrorSourceFlexRayFilter
BSW Parameter	BSW Type
MirrorSourceFlexRayFilterUpperSlot	EcucIntegerParamDef
BSW Description	
Highest slot ID that is accepted by the filter.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00045]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay
BSW Parameter	BSW Type
MirrorSourceMaxDynamicFilters	EcucIntegerParamDef
BSW Description	
Maximum number of filters that can be dynamically added using Mirror_AddXxxFilter().	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00013]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork
BSW Parameter	BSW Type
MirrorSourceNetworkLin	EcucParamConfContainerDef
BSW Description	
Source bus representing a LIN network.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannel.channel	
Mapping Rule	Mapping Type
Create a container for each LinPhysicalChannel which is available in the System Extract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00029]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	BSW Type
MirrorComMNetworkHandleRef	EcucReferenceDef
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannel.channel	
Mapping Rule	Mapping Type

<p>This reference shall be derived from the:</p> <ul style="list-style-type: none"> - CanCluster that aggregates the AbstractCanPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkFlexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworklp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container 		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00064]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin	
BSW Parameter		BSW Type
MirrorNetworkId		EcucIntegerParamDef
BSW Description		
Network ID of the bus.		
Template Description		
This attribute defines the networkId of the communication channel.		
M2 Parameter		
SystemTemplate::BusMirror::BusMirrorChannel.busMirrorNetworkId		
Mapping Rule		Mapping Type
<p>If this parameter is aggregated by MirrorDestNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingCan.</p> <p>If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingFlexray.</p> <p>If this parameter is aggregated by MirrorDestNetworklp container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappinglp.</p> <p>If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the BusMirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMappingUserDefined.</p> <p>If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p> <p>If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannelMapping.</p>		full
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Mirror_00012]
-------	---------------------

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin	
BSW Parameter	BSW Type	
MirrorSourceLinFilter	EcucChoiceContainerDef	
BSW Description	Pre-configured filter for LIN frames.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00030]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter	
BSW Parameter	BSW Type	
MirrorSourceLinFilterMask	EcucParamConfContainerDef	
BSW Description	Pre-configured mask based filter for LIN frames.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00035]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterMask	
BSW Parameter	BSW Type	
MirrorSourceLinFilterId	EcucIntegerParamDef	
BSW Description	Unique identifier of the pre-configured LIN filter.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00034]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterMask	
BSW Parameter	BSW Type	
MirrorSourceLinFilterLinIdCode	EcucIntegerParamDef	
BSW Description	Value to match masked frame IDs.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00036]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterMask	
BSW Parameter	BSW Type	
MirrorSourceLinFilterLinIdMask	EcucIntegerParamDef	
BSW Description	Mask applied to frame IDs before comparison.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00037]	

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter	
BSW Parameter	BSW Type	
MirrorSourceLinFilterRange	EcucParamConfContainerDef	
BSW Description	Pre-configured range filter for LIN frames.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Mirror_00031]	

BSW Module	BSW Context
------------	-------------

Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterRange	
BSW Parameter		BSW Type
MirrorSourceLinFilterId	EcucIntegerParamDef	
BSW Description		
Unique identifier of the pre-configured LIN filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00034]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterRange	
BSW Parameter		BSW Type
MirrorSourceLinFilterLower	EcucIntegerParamDef	
BSW Description		
Lowest frame ID that is accepted by the filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00032]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinFilter/MirrorSourceLinFilterRange	
BSW Parameter		BSW Type
MirrorSourceLinFilterUpper	EcucIntegerParamDef	
BSW Description		
Highest frame ID that is accepted by the filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00033]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin	
BSW Parameter		BSW Type

MirrorSourceLinToCanBaselId	EcucIntegerParamDef
BSW Description	
Base ID merged with the LIN frame ID to form the CAN ID.	
Template Description	
Base ID merged with the LIN frame ID to form the CAN ID.	
Only required when a BusMirrorChannel that refers to a LinPhysicalChannel in the role channel is referenced in the role sourceChannel.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorChannelMappingCan.mirrorSourceLinToCanRangeBaselId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00041]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	BSW Type
MirrorSourceLinToCanIdMapping	EcucParamConfContainerDef
BSW Description	
Rule for mapping a LIN frame ID to a special CAN ID.	
Template Description	
This element defines a rule for remapping a single LIN Frame.	
M2 Parameter	
SystemTemplate::BusMirror::BusMirrorLinPidToCanIdMapping	
Mapping Rule	Mapping Type
Create container in case that the BusMirrorLinPidToCanIdMapping is aggregated by BusMirrorChannelMappingCan in the role linPidToCanIdMapping.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00038]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinToCanIdMapping
BSW Parameter	BSW Type
MirrorSourceLinToCanIdMappingCanId	EcucIntegerParamDef
BSW Description	
CAN ID which lies outside of the range mapping.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00040]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin/MirrorSourceLinToCanIdMapping

BSW Parameter		BSW Type
MirrorSourceLinToCanIdMappingLinId		EcucIntegerParamDef
BSW Description		
Frame ID which is excluded from the range mapping.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00039]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin	
BSW Parameter		BSW Type
MirrorSourceMaxDynamicFilters		EcucIntegerParamDef
BSW Description		
Maximum number of filters that can be dynamically added using Mirror_AddXxxFilter().		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00013]

BSW Module	BSW Context	
Mirror	Mirror	
BSW Parameter		BSW Type
MirrorGeneral		EcucParamConfContainerDef
BSW Description		
Contains the general configuration parameters of the module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Mirror_00002]

BSW Module	BSW Context	
Mirror	Mirror/MirrorGeneral	
BSW Parameter		BSW Type
MirrorDevErrorDetect		EcucBooleanParamDef
BSW Description		

Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00003]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral
BSW Parameter	BSW Type
MirrorEcucPartitionRef	EcucReferenceDef
BSW Description	
Reference to EcucPartition, where BusMirroring module is assigned to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00067]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral
BSW Parameter	BSW Type
MirrorMainFunction	EcucParamConfContainerDef
BSW Description	
Each element of this container defines one instance of Mirror_MainFunction.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00068]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral/MirrorMainFunction
BSW Parameter	BSW Type
MirrorMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Execution cycle of the respective Mirror_MainFunction instance in seconds.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00070]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral/MirrorMainFunction
BSW Parameter	BSW Type
MirrorMainPartitionRef	EcucReferenceDef
BSW Description	
Reference to EcucPartition, where the according Mirror_MainFunction instance is assigned to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00069]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral
BSW Parameter	BSW Type
MirrorStbRef	EcucReferenceDef
BSW Description	
Reference to the StbM time base to use for acquiring the time stamps used in the mirroring protocol.	
This reference is not required if all destination buses are CAN.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00065]

BSW Module	BSW Context
Mirror	Mirror/MirrorGeneral
BSW Parameter	BSW Type
MirrorVersionInfoApi	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling version info API support.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00005]

C.2 Can

C.2.1 Can Driver Mapping

BSW Module	BSW Context
Can	Can
BSW Parameter	BSW Type
CanConfigSet	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR Can module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00343]

BSW Module	BSW Context
Can	Can/CanConfigSet
BSW Parameter	BSW Type
CanController	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the CAN controller(s).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00354]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController
BSW Parameter	BSW Type
CanBusoffProcessing	EcucEnumerationParamDef
BSW Description	
Enables / disables API Can_MainFunction_BusOff() for handling busoff events in polling mode.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00314]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanControllerActivation		EcucBooleanParamDef
BSW Description		
Defines if a CAN controller is used in the configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00315]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanControllerBaseAddress		EcucIntegerParamDef
BSW Description		
Specifies the CAN controller base address.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00382]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanControllerBaudrateConfig		EcucParamConfContainerDef
BSW Description		
This container contains bit timing related configuration parameters of the CAN controller(s).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00387]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter	BSW Type	
CanControllerBaudRate	EcucFloatParamDef	
BSW Description		
Specifies the baudrate of the controller in kbps.		
Template Description		
Channels speed in bits/s.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		
Mapping Rule		Mapping Type
SystemTemplate speed is in bps, so divide it by 1000 to get kbps		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00005]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter	BSW Type	
CanControllerBaudRateConfigID	EcucIntegerParamDef	
BSW Description		
Uniquely identifies a specific baud rate configuration. This ID is used by SetBaudrate API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00471]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter	BSW Type	
CanControllerFdBaudrateConfig	EcucParamConfContainerDef	
BSW Description		
This optional container contains bit timing related configuration parameters of the CAN controller(s) for payload and CRC of a CAN FD frame. If this container exists the controller supports CAN FD frames.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00473]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	

BSW Parameter		BSW Type
CanControllerFdBaudRate		EcucFloatParamDef
BSW Description		
Specifies the data segment baud rate of the controller in kbps.		
Template Description		
Specifies the data segment baud rate of the controller in bits/s.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::AbstractCanCluster.canFdBaudrate		
Mapping Rule		Mapping Type
SystemTemplate speed is in bps, so divide it by 1000 to get kbps		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00481]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		BSW Type
CanControllerPropSeg		EcucIntegerParamDef
BSW Description		
Specifies propagation delay in time quantas.		
Template Description		
Specifies propagation delay in time quantas.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.propSeg		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00476]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		BSW Type
CanControllerSeg1		EcucIntegerParamDef
BSW Description		
Specifies phase segment 1 in time quantas.		
Template Description		
Specifies phase segment 1 in time quantas.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.timeSeg1		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00477]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		BSW Type
CanControllerSeg2		EcucIntegerParamDef

BSW Description	
Specifies phase segment 2 in time quantas.	
Template Description	
Specifies phase segment 2 in time quantas.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.timeSeg2	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00478]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	BSW Type
CanControllerSspOffset	EcucIntegerParamDef
BSW Description	
<p>Specifies the Transmitter Delay Compensation Offset in minimum time quanta (see [17]). Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified, Transmitter Delay Compensation is disabled.</p> <p>Note: MTQ == Minimum Time Quanta in seconds == $1/(\text{frequency of the CAN controller clock})$ Secondary Sample Point Offset in seconds = $\text{CanControllerSspOffset} * \text{MTQ}$</p> <p>Example: CAN controller clock frequency = 20MHz => $\text{MTQ} = 1/20 * 10^(-6) \text{ s} = 0,05 \text{ us} = 50\text{ns}$ Baud rate = 1MBit/s => $\text{BitTime} = 1/(1 * 10^6) \text{ s/Bit} = 1 * 10^(-6) = 1\text{us/Bit}$ SSP = 75% => $\text{SSP in seconds} = 0,75 * 1\text{us} = 750 \text{ ns}$ $\text{CanControllerSspOffset in MTQ} = 750\text{ns} / 50\text{ns} = 15$</p> <p>Note: Please consider the minimum range (0..63) stated in [17] and the range definition (0..127) used as per [19].</p>	
Template Description	
Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.sspOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00494]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	BSW Type

CanControllerSyncJumpWidth	EcucIntegerParamDef
BSW Description	
Specifies the synchronization jump width for the controller in time quantas.	
Template Description	
Specifies the synchronization jump width for the controller in time quantas.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.syncJumpWidth	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00479]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	BSW Type
CanControllerTxBitRateSwitch	EcucBooleanParamDef
BSW Description	
Specifies if the bit rate switching shall be used for transmissions. If FALSE: CAN FD frames shall be sent without bit rate switching.	
Template Description	
Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.txBitRateSwitch	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00475]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	BSW Type
CanControllerPropSeg	EcucIntegerParamDef
BSW Description	
Specifies propagation delay in time quantas.	
Template Description	
Specifies propagation delay in time quantas.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.propSeg	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00073]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig

BSW Parameter		BSW Type	
CanControllerSeg1		EcucIntegerParamDef	
BSW Description			
Specifies phase segment 1 in time quantas.			
Template Description			
Specifies phase segment 1 in time quantas.			
timeSeg1 = Phase_Seg1			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.timeSeg1			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00074]	

BSW Module	BSW Context		
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig		
BSW Parameter		BSW Type	
CanControllerSeg2		EcucIntegerParamDef	
BSW Description			
Specifies phase segment 2 in time quantas.			
Template Description			
Specifies phase segment 2 in time quantas.			
timeSeg2 = Phase_Seg2			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.timeSeg2			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00075]	

BSW Module	BSW Context		
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig		
BSW Parameter		BSW Type	
CanControllerSyncJumpWidth		EcucIntegerParamDef	
BSW Description			
Specifies the synchronization jump width for the controller in time quantas.			
Template Description			
The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.syncJumpWidth			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00383]	

BSW Module	BSW Context
------------	-------------

Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanControllerDefaultBaudrate		EcucReferenceDef
BSW Description		
Reference to baudrate configuration container configured for the Can Controller.		
Template Description		
Channels speed in bits/s.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		
Mapping Rule		Mapping Type
Set the reference to the container of the CanControllerBaudRate parameter that has been configured for SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00435]

BSW Module		BSW Context
Can		Can/CanConfigSet/CanController
BSW Parameter		BSW Type
CanControllerEcucPartitionRef		EcucReferenceDef
BSW Description		
Maps the CAN controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the CAN driver is mapped to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00492]

BSW Module		BSW Context
Can		Can/CanConfigSet/CanController
BSW Parameter		BSW Type
CanControllerId		EcucIntegerParamDef
BSW Description		
This parameter provides the controller ID which is unique in a given CAN Driver. The value for this parameter starts with 0 and continue without any gaps.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00316]

BSW Module		BSW Context
Can		Can/CanConfigSet/CanController

BSW Parameter		BSW Type	
CanCpuClockRef		EcucReferenceDef	
BSW Description			
Reference to the CPU clock configuration, which is set in the MCU driver configuration			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00313]	

BSW Module		BSW Context	
Can		Can/CanConfigSet/CanController	
BSW Parameter		BSW Type	
CanRxProcessing		EcucEnumerationParamDef	
BSW Description			
Enables / disables API Can_MainFunction_Read() for handling PDU reception events in polling mode.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00317]	

BSW Module		BSW Context	
Can		Can/CanConfigSet/CanController	
BSW Parameter		BSW Type	
CanTTController		EcucParamConfContainerDef	
BSW Description			
CanTTController is specified in the SWS TTCAN and contains the configuration parameters of the TTCAN controller(s) (which are needed in addition to the configuration parameters of the CAN controller(s)).			
This container is only included and valid if TTCAN is supported by the controller, enabled (see CanSupportTTCANRef, ECUC_Can_00430), and used.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00001]	

BSW Module		BSW Context	

Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		BSW Type
CanTTControllerApplWatchdogLimit		EcucIntegerParamDef
BSW Description		
Defines the maximum time period (unit is 256 times NTU) after which the application has to serve the watchdog.		
Template Description		
The Appl_Watchdog_Limit shall be an 8-bit value specifying the period for the application watchdog in Appl_Watchdog_Limit times 256 NTUs.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.applWatchdogLimit		
Mapping Rule		Mapping Type
1:1 mapping		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00139]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		BSW Type
CanTTControllerCycleCountMax		EcucIntegerParamDef
BSW Description		
Defines the value for cycle_count_max. Allowed values: 0x00: 1 basic cycle 0x01: 2 basic cycles 0x03: 4 basic cycles 0x07: 8 basic cycles 0x0F: 16 basic cycles 0x1F: 32 basic cycles 0x3F: 64 basic cycles		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00138]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		BSW Type
CanTTControllerEcucPartitionRef		EcucReferenceDef
BSW Description		
Maps the Time triggered CAN controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the CAN driver is mapped to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00493]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerExpectedTxTrigger	EcucIntegerParamDef	
BSW Description		
Number of expected_tx_trigger.		
Template Description		
The Expected_Tx_Trigger shall be an eight (8) bit value which limits the number of messages the FSE may try to transmit in one matrix cycle.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.expectedTxTrigger		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00136]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerExternalClockSynchronisation	EcucBooleanParamDef	
BSW Description		
Enables/disables the external clock synchronization. TRUE: External clock synchronization enabled. FALSE: External clock synchronization disabled.		
This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
Template Description		
One bit shall be used to configure whether or not external clock synchronisation will be allowed during runtime (only Level 2).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.externalClockSynchronisation		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00135]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerGlobalTimeFiltering	EcucBooleanParamDef	
BSW Description		

Enables/disables the global time filtering. TRUE: Global time filtering enabled. FALSE: Global time filtering disabled.	
This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00134]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type
CanTTControllerInitialRefOffset	EcucIntegerParamDef
BSW Description	
Defines the initial value for ref trigger offset.	
Template Description	
The Initial_Ref_Offset shall be an eight (8) bit value for the initialisation of Ref_Trigger_Offset.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.initialRefOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00128]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type
CanTTControllerInterruptEnable	EcucIntegerParamDef
BSW Description	

<p>Enables/disables the respective interrupts. Bit Position set to 1: Enable respective interrupt. Bit Position set to 0: Disable respective interrupt.</p> <p>Bit Position / Interrupt Source: 10: Application Watchdog. 9: Watch Trigger reached. 8: Initialization Watch Trigger reached. 7: Change of Error Level. 6: Tx Overflow. 5: Tx Underflow. 4: Global Time Error. 3: Gap. 2: Start of Cycle. 1: Time Discontinuity. 0: Master State Change.</p> <p>Bit position "1: Time Discontinuity" and "4: Global Time Error" shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00140]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type
CanTTControllerLevel2	EcucBooleanParamDef
BSW Description	
<p>Defines whether Level 2 or Level 1 is used. TRUE: Level 2. FALSE: Level 1.</p> <p>If this parameter is set to FALSE then all parameters with dependency to CanTTControllerLevel2 need not be configured.</p>	
Template Description	
One bit shall be used to distinguish between Level 1 and Level 2.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.timeTriggeredCanLevel	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00131]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type

CanTTControllerNTUConfig	EcucFloatParamDef
BSW Description	
<p>Defines the config value for NTU (network time unit). Value given in microseconds. The value configured shall be greater than 0. Together with the local oscillator period, the TUR (time unit ratio) can be derived from the NTU. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.</p>	
Template Description	
Unit measuring all times and providing a constant of the whole network. For level 1, this is always the CAN bit time. Unit: seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCluster.ntu	
Mapping Rule	Mapping Type
NTU = system clock period x (TUR Numerator / TUR Denominator)	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00141]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type
CanTTControllerOperationMode	EcucEnumerationParamDef
BSW Description	
Defines the operation mode.	
Template Description	
Possible operation modes	
True: Time-Triggered	
False: Event-Synchronised-Time-Triggered	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCluster.operationMode	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00127]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanTTController
BSW Parameter	BSW Type
CanTTControllerSyncDeviation	EcucFloatParamDef
BSW Description	
<p>Defines the maximum synchronization deviation: Given as a percentage value of the NTU (network time unit). The value configured shall be greater than 0. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Synchronisation Deviation $\leq 2^{\wedge} (\text{CanTTSyncDeviation} + 5)$.	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00132]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerTURRestore	EcucBooleanParamDef	
BSW Description		
Enables/disables the TUR restore. Note that the value configured for TUR can be derived from the value configured for NTU and the local oscillator period. TRUE: TUR restore enabled. FALSE: TUR restore disabled. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00133]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerTimeMaster	EcucBooleanParamDef	
BSW Description		
Defines whether the controller acts as a potential time master. TRUE: Potential time master. FALSE: Time slave.		
Template Description		
One bit shall be used to distinguish between (potential) time masters and time slaves. This can be derived from the frame-triggering's triggers.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.master		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00129]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerTimeMasterPriority	EcucIntegerParamDef	
BSW Description		
Defines the time master priority.		
Template Description		
The time master priority shall contain a three bit value for the priority of the current time master (the last three bits of the identifier of the reference message). This can be derived from the frame-triggering's triggers.		
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.timeMaster Priority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00130]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerTxEnableWindowLength	EcucIntegerParamDef	
BSW Description		
Length of the tx enable window given in CAN bit times. Definition parameter "CanTTControllerTxEnableWindowlength" is used such that: Length of enable window = CanTTControllerTxEnableWindowLength + 1		
Template Description		
The length of the Tx_Enable window shall be a four (4) bit value specifying the length of the time period (1-16 nominal CAN bit times) in which a transmission may be started.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.txEnable WindowLength		
Mapping Rule	Mapping Type	
Length of enable window = CanTTControllerTxEnableWindowLength + 1	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00137]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerWatchTriggerGapTimeMark	EcucIntegerParamDef	
BSW Description		
watch trigger time mark after a gap		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00158]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter	BSW Type	
CanTTControllerWatchTriggerTimeMark	EcucIntegerParamDef	
BSW Description		
watch trigger time mark		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00157]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		BSW Type
CanTTIRQProcessing		EcucEnumerationParamDef
BSW Description		
Enables / disables API Can_MainFunction_BusOff() for handling busoff events in polling mode.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00142]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanTxProcessing		EcucEnumerationParamDef
BSW Description		
Enables / disables API Can_MainFunction_Write() for handling PDU transmission events in polling mode.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00318]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanWakeupFunctionalityAPI		EcucBooleanParamDef
BSW Description		
Adds / removes the service Can_CheckWakeup() from the code. True: Can_CheckWakeup can be used. False: Can_CheckWakeup cannot be used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00466]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanWakeupProcessing		EcucEnumerationParamDef
BSW Description		
Enables / disables API Can_MainFunction_Wakeup() for handling wakeup events in polling mode.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00319]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanWakeupSourceRef		EcucReferenceDef
BSW Description		
This parameter contains a reference to the Wakeup Source for this controller as defined in the ECU State Manager.		
Implementation Type: reference to EcuM_WakeupSourceType		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00359]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		BSW Type
CanWakeupSupport		EcucBooleanParamDef
BSW Description		
CAN driver support for wakeup over CAN Bus.		
Template Description		

Defines whether the ECU shall be woken up by this CommunicationController.	
TRUE: wake up is possible	
FALSE: wake up is not supported	
Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationController.wakeUpByControllerSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00330]

BSW Module	BSW Context
Can	Can/CanConfigSet
BSW Parameter	BSW Type
CanHardwareObject	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of CAN Hardware Objects.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00324]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanControllerRef	EcucReferenceDef
BSW Description	
Reference to CAN Controller to which the HOH is associated to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00322]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanFdPaddingValue	EcucIntegerParamDef

BSW Description	
Specifies the value which is used to pad unspecified data in CAN FD frames > 8 bytes for transmission. This is necessary due to the discrete possible values of the DLC if > 8 bytes.	
If the length of a PDU which was requested to be sent does not match the allowed DLC values, the remaining bytes up to the next possible value shall be padded with this value.	
Template Description	
CanControllerFdConfiguration.paddingValue: Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.	
CanControllerFdConfigurationRequirements.paddingValue: Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.paddingValue, SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfigurationRequirements.paddingValue	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00485]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanHandleType	EcucEnumerationParamDef
BSW Description	
Specifies the type (Full-CAN or Basic-CAN) of a hardware object.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00323]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanHardwareObjectUsesPolling	EcucBooleanParamDef
BSW Description	
Enables polling of this hardware object.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00490]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter	BSW Type	
CanHwFilter	EcucParamConfContainerDef	
BSW Description		
This container is only valid for HRHs and contains the configuration (parameters) of one hardware filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00468]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanHwFilter	
BSW Parameter	BSW Type	
CanHwFilterCode	EcucIntegerParamDef	
BSW Description		
Specifies (together with the filter mask) the identifiers range that passes the hardware filter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00469]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanHwFilter	
BSW Parameter	BSW Type	
CanHwFilterMask	EcucIntegerParamDef	
BSW Description		
Describes a mask for hardware-based filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with the appropriate CanFilterMaskValue. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position.		
The mask shall be build by filling with leading 0. In case of CanIdType EXTENDED or MIXED a 29 bit mask shall be build. In case of CanIdType STANDARD a 11 bit mask shall be build		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00470]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		BSW Type
CanHwObjectCount		EcucIntegerParamDef
BSW Description		
Number of hardware objects used to implement one HOH. In case of a HRH this parameter defines the number of elements in the hardware FIFO or the number of shadow buffers, in case of a HTH it defines the number of hardware objects used for multiplexed transmission or for a hardware FIFO used by a FullCAN HTH.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00467]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		BSW Type
CanIdType		EcucEnumerationParamDef
BSW Description		
Specifies whether the IdValue is of type		
<ul style="list-style-type: none"> - standard identifier - extended identifier - mixed mode 		
ImplementationType: Can_IdType		
Template Description		
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00065]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanIdType	
BSW Parameter		BSW Type
EXTENDED		EcucEnumerationLiteralDef

BSW Description	
All the CANIDs are of type extended only (29 bit).	
Template Description	
Extended 29-bit-identifiers are used (CAN 2.0B)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanAddressingModeType.extended	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanIdType
BSW Parameter	BSW Type
STANDARD	EcucEnumerationLiteralDef
BSW Description	
All the CANIDs are of type standard only (11bit).	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanAddressingModeType.standard	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanMainFunctionRWPeriodRef	EcucReferenceDef
BSW Description	
Reference to CanMainFunctionPeriod	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00438]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanObjectId	EcucIntegerParamDef
BSW Description	

<p>Holds the handle ID of HRH or HTH. The value of this parameter is unique in a given CAN Driver, and it should start with 0 and continue without any gaps.</p> <p>The HRH and HTH Ids share a common ID range.</p> <p>Example: HRH0-0, HRH1-1, HTH0-2, HTH1-3</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00326]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanObjectType	EcucEnumerationParamDef
BSW Description	
Specifies if the HardwareObject is used as Transmit or as Receive object	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00327]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	BSW Type
CanTTHardwareObjectTrigger	EcucParamConfContainerDef
BSW Description	
<p>CanTTHardwareObjectTrigger is specified in the SWS TTCAN and contains the configuration (parameters) of TTCAN triggers for Hardware Objects, which are additional to the configuration (parameters) of CAN Hardware Objects.</p> <p>This container is only included and valid if TTCAN is supported by the controller and, enabled (see CanSupportTTCANRef, ECUC_Can_00430), and used.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00002]

BSW Module	BSW Context
-------------------	--------------------

Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		BSW Type
CanTTHardwareObjectBaseCycle		EcucIntegerParamDef
BSW Description		
Defines the cycle_offset. CanTTHardwareObjectBaseCycle must be not greater than cycle_count_max.		
Template Description		
The first communication cycle where the frame is sent.		
This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.BaseCycle		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00147]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		BSW Type
CanTTHardwareObjectCycleRepetition		EcucIntegerParamDef
BSW Description		
Defines the repeat_factor.		
CanTTHardwareObjectCycleRepetition shall be a power of two (2), greater than cycle_offset but not greater than cycle_count_max + 1.		
Template Description		
The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.CycleRepetition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00148]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		BSW Type
CanTTHardwareObjectTimeMark		EcucIntegerParamDef
BSW Description		
Defines the point in time, when the trigger will be activated. Value is given in cycle time.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00146]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter	BSW Type	
CanTTHardwareObjectTriggerId	EcucIntegerParamDef	
BSW Description	Sequential number which allows separation of different TTCAN triggers configured for one and the same hardware object.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00155]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter	BSW Type	
CanTTHardwareObjectTriggerType	EcucEnumerationParamDef	
BSW Description	Defines the type of the trigger associated with the hardware object. This parameter depends on plain CAN parameter CAN_OBJECT_TYPE. If CAN_OBJECT_TYPE equals RECEIVE than this parameter is fixed to CAN_TT_RX_TRIGGER. If CAN_OBJECT_TYPE equals TRANSMIT than one of the following literals is configurable: CAN_TT_TX_REF_TRIGGER, CAN_TT_TX_REF_TRIGGER_GAP, CAN_TT_TX_TRIGGER_MERGED, CAN_TT_TX_TRIGGER_SINGLE, CAN_TT_TX_TRIGGER_EXCLUSIVE.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00145]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter	BSW Type	
CanTriggerTransmitEnable	EcucBooleanParamDef	
BSW Description	This parameter defines if or if not Can supports the trigger-transmit API for this handle.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	

Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00486]

BSW Module	BSW Context	
Can	Can/CanConfigSet	
BSW Parameter	BSW Type	
Canlcom	EcucParamConfContainerDef	
BSW Description		
This container contains the parameters for configuring pretended networking		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00440]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom	
BSW Parameter	BSW Type	
CanlcomConfig	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters of the ICOM Configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00459]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig	
BSW Parameter	BSW Type	
CanlcomConfigId	EcucIntegerParamDef	
BSW Description		
This parameter identifies the ID of the ICOM configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00441]	

BSW Module	BSW Context

Can	Can/CanConfigSet/Canlcom/CanlcomConfig	
BSW Parameter		BSW Type
CanlcomWakeOnBusOff		EcucBooleanParamDef
BSW Description		
This parameter defines that the MCU shall wake if the bus off is detected or not.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00442]

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig	
BSW Parameter		BSW Type
CanlcomWakeupCauses		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the wakeup causes to leave the power saving mode.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00443]

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses	
BSW Parameter		BSW Type
CanlcomRxMessage		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters for the wakeup causes for matching received messages. It has to be configured as often as received messages are defined as wakeup cause.		
constraint: For all CanlcomRxMessage instances the Message IDs which are defined in CanlcomMessageld and in CanlcomRxMessageldMask shall not overlap.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00447]

BSW Module	BSW Context
-------------------	--------------------

Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter		BSW Type
CanlcomCounterValue		EcucIntegerParamDef
BSW Description		
This parameter defines that the MCU shall wake if the message with the ID is received n times on the communication channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00448]

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter		BSW Type
CanlcomMessageld		EcucIntegerParamDef
BSW Description		
This parameter defines the message ID the wakeup causes of this CanlcomRxMessage are configured for. In addition a mask (CanlcomMessageldMask) can be defined, in that case it is possible to define a range of rx messages, which can create a wakeup condition.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00449]

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter		BSW Type
CanlcomMessageldMask		EcucIntegerParamDef
BSW Description		
Describes a mask for filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with this CanlcomMessageldMask. If the masked identifier matches the masked value of CanlcomMessageld, it can create a wakeup condition for this CanlcomRxMessage. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position. The mask shall be build by filling with leading 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID

valid	[ECUC_Can_00465]
-------	------------------

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter	BSW Type	
CanlcomMissingMessageTimerValue	EcucFloatParamDef	
BSW Description		
This parameter defines that the MCU shall wake if the message with the ID is not received for a specific time in s on the communication channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00450]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter	BSW Type	
CanlcomPayloadLengthError	EcucBooleanParamDef	
BSW Description		
This parameter defines that the MCU shall wake if a payload error occurs		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00451]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage	
BSW Parameter	BSW Type	
CanlcomRxMessageSignalConfig	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters for the wakeup causes for matching signals. It has to be configured as often as a signal is defined as wakeup cause. If at least one Signal conditions defined in a CanlcomRxMessageSignalConfig evaluates to true or if no CanlcomRxMessageSignalConfig are defined, the whole wakeup condition is considered to be true. All instances of this container refer to the same frame/pdu (see CanlcomMessageId).		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00452]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter	BSW Type	
CanIcomSignalMask	EcucIntegerParamDef	
BSW Description		
This parameter shall be used to mask a signal in the payload of a CAN message. The mask is binary AND with the signal payload. The result will be used in combination of the operations defined in CanIcomSignalOperation with the CanIcomSignalValue.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00487]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter	BSW Type	
CanIcomSignalOperation	EcucEnumerationParamDef	
BSW Description		
This parameter defines the operation, which shall be used to verify the signal value creates a wakeup condition.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00462]	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter	BSW Type	
CanIcomSignalRef	EcucReferenceDef	
BSW Description		
This parameter defines a reference to the signal which shall be checked additional to the message id (CanIcomMessageId). This reference is used for documentation to define which ComSignal originates this filter setting. All signals being referred by this reference shall point to the same PDU.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00456]

BSW Module	BSW Context
Can	Can/CanConfigSet/Canlcom/CanlcomConfig/CanlcomWakeupCauses/CanlcomRxMessage/CanlcomRxMessageSignalConfig
BSW Parameter	BSW Type
CanlcomSignalValue	EcucIntegerParamDef
BSW Description	
This parameter shall be used to define a signal value which shall be compared (CanlcomSignalOperation) with the masked CanlcomSignalMask value of the received signal (CanlcomSignalRef).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00488]

BSW Module	BSW Context
Can	Can
BSW Parameter	BSW Type
CanGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the parameters related each CAN Driver Unit.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00328]

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	BSW Type
CanDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00064]

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	BSW Type
CanEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the CAN driver to zero or multiple ECUC partitions to make the modules API available in this partition. The CAN driver will operate as an independent instance in each of the partitions.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00491]

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	BSW Type
CanIcomGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the ICOM Configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00444]

BSW Module	BSW Context
Can	Can/CanGeneral/CanIcomGeneral
BSW Parameter	BSW Type
CanIcomLevel	EcucEnumerationParamDef
BSW Description	
Defines the level of Pretended Networking. This parameter is reserved for future implementations (Pretended Networking level 2).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00445]

BSW Module	BSW Context	
Can	Can/CanGeneral/CanIcomGeneral	
BSW Parameter	BSW Type	
CanIcomVariant	EcucEnumerationParamDef	
BSW Description		
Defines the variant, which is supported by this CanController		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00446]	

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter	BSW Type	
CanIndex	EcucIntegerParamDef	
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Can_00320]	

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter	BSW Type	
CanLPduReceiveCalloutFunction	EcucFunctionNameDef	
BSW Description		
This parameter defines the existence and the name of a callout function that is called after a successful reception of a received CAN Rx L-PDU. If this parameter is omitted no callout shall take place.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Can_00434]
-------	------------------

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter	BSW Type	
CanMainFunctionBusoffPeriod	EcucFloatParamDef	
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Busoff. Unit is seconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00355]

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter	BSW Type	
CanMainFunctionModePeriod	EcucFloatParamDef	
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Mode. Unit is seconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00376]

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter	BSW Type	
CanMainFunctionRWPeriods	EcucParamConfContainerDef	
BSW Description		
This container contains the parameter for configuring the period for cyclic call to Can_MainFunction_Read or Can_MainFunction_Write depending on the referring item.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00437]

BSW Module	BSW Context

Can	Can/CanGeneral/CanMainFunctionRWPeriods	
BSW Parameter		BSW Type
CanMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Read or Can_MainFunction_Write depending on the referring item. Unit is seconds. Different poll-cycles will be configurable if more than one CanMainFunctionPeriod is configured. In this case multiple Can_MainFunction_Read() or Can_MainFunction_Write() will be provided by the CAN Driver module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00484]

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter		BSW Type
CanMainFunctionWakeupPeriod	EcucFloatParamDef	
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Wakeup. Unit is seconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00357]

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter		BSW Type
CanMultiplexedTransmission	EcucBooleanParamDef	
BSW Description		
Specifies if multiplexed transmission shall be supported.ON or OFF		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Can_00095]

BSW Module	BSW Context	
Can	Can/CanGeneral	

BSW Parameter		BSW Type	
CanOsCounterRef		EcucReferenceDef	
BSW Description			
This parameter contains a reference to the OsCounter, which is used by the CAN driver.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00431]	

BSW Module		BSW Context	
Can		Can/CanGeneral	
BSW Parameter		BSW Type	
CanPublicIcomSupport		EcucBooleanParamDef	
BSW Description			
Selects support of Pretended Network features in Can driver. True: Enabled False: Disabled			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00483]	

BSW Module		BSW Context	
Can		Can/CanGeneral	
BSW Parameter		BSW Type	
CanSetBaudrateApi		EcucBooleanParamDef	
BSW Description			
The support of the Can_SetBaudrate API is optional. If this parameter is set to true the Can_SetBaudrate API shall be supported. Otherwise the API is not supported.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Can_00482]	

BSW Module		BSW Context	
Can		Can/CanGeneral	
BSW Parameter		BSW Type	

CanSupportTTCANRef	EcucReferenceDef
BSW Description	
The parameter refers to CanIfSupportTTCAN parameter in the CAN Interface Module configuration.	
The CanIfSupportTTCAN parameter defines whether TTCAN is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00430]

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	BSW Type
CanTimeoutDuration	EcucFloatParamDef
BSW Description	
Specifies the maximum time for blocking function until a timeout is detected. Unit is seconds.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00113]

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	BSW Type
CanVersionInfoApi	EcucBooleanParamDef
BSW Description	
Switches the Can_GetVersionInfo() API ON or OFF.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Can_00106]

C.2.2 Can Interface Mapping

BSW Module	BSW Context
CanIf	CanIf
BSW Parameter	BSW Type
CanIfCtrlDrvCfg	EcucParamConfContainerDef

BSW Description	
Configuration parameters for all the underlying CAN Driver modules are aggregated under this container. For each CAN Driver module a separate instance of this container has to be provided.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00253]

BSW Module	BSW Context
CanIf	CanIf/CanIfCtrlDrvCfg
BSW Parameter	BSW Type
CanIfCtrlCfg	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of an addressed CAN controller by an underlying CAN Driver module. This container is configurable per CAN controller.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00546]

BSW Module	BSW Context
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg
BSW Parameter	BSW Type
CanIfCtrlCanCtrlRef	EcucReferenceDef
BSW Description	
This parameter references to the logical handle of the underlying CAN controller from the CAN Driver module to be served by the CAN Interface module. The following parameters of CanController config container shall be referenced by this link: CanControllerId, CanWakeupSourceRef	
Range: 0..max. number of underlying supported CAN controllers	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00636]

BSW Module	BSW Context
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg
BSW Parameter	BSW Type

CanIfCtrlId	EcucIntegerParamDef
BSW Description	
This parameter abstracts from the CAN Driver specific parameter Controller. Each controller of all connected CAN Driver modules shall be assigned to one specific ControllerId of the CanIf. Range: 0..number of configured controllers of all CAN Driver modules	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	
valid	ECUC Parameter ID [ECUC_CanIf_00647]

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg	
BSW Parameter		BSW Type
CanIfCtrlWakeupSupport		EcucBooleanParamDef
BSW Description		
This parameter defines if a respective controller of the referenced CAN Driver modules is queriable for wake up events. True: Enabled False: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		
	Mapping Type local	
Mapping Status		
valid	ECUC Parameter ID [ECUC_CanIf_00637]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg	
BSW Parameter		BSW Type
CanIfCtrlDrvInitHohConfigRef		EcucReferenceDef
BSW Description		
Reference to the Init Hoh Configuration		
Template Description		
M2 Parameter		
Mapping Rule		
	Mapping Type local	
Mapping Status		
valid	ECUC Parameter ID [ECUC_CanIf_00642]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg	

BSW Parameter		BSW Type	
CanIfCtrlDrvNameRef		EcucReferenceDef	
BSW Description			
CAN Interface Driver Reference.			
This reference can be used to get any information (Ex. Driver Name, Vendor ID) from the CAN driver.			
The CAN Driver name can be derived from the ShortName of the CAN driver module.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00638]	

BSW Module		BSW Context	
CanIf		CanIf	
BSW Parameter		BSW Type	
CanIfDispatchCfg		EcucParamConfContainerDef	
BSW Description			
Callback functions provided by upper layer modules of the CanIf. The callback functions defined in this container are common to all configured CAN Driver / CAN Transceiver Driver modules.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00250]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserCheckTrcvWakeFlagIndicationName		EcucFunctionNameDef	
BSW Description			
This parameter defines the name of <User_CheckTrcvWakeFlagIndication>.			
If CanIfDispatchUserCheckTrcvWakeFlagIndicationUL equals CAN_SM the name of <User_CheckTrcvWakeFlagIndication> is fixed. If it equals CDD, the name is selectable. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00791]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserCheckTrcvWakeFlagIndicationUL		EcucEnumerationParamDef	
BSW Description			
This parameter defines the upper layer module to which the CheckTrcvWakeFlagIndication from the Driver modules have to be routed. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00792]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserClearTrcvWufFlagIndicationName		EcucFunctionNameDef	
BSW Description			
This parameter defines the name of <User_ClearTrcvWufFlagIndication>. If CanIfDispatchUserClearTrcvWufFlagIndicationUL equals CAN_SM the name of <User_ClearTrcvWufFlagIndication> is fixed. If it equals CDD, the name is selectable. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00789]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserClearTrcvWufFlagIndicationUL		EcucEnumerationParamDef	
BSW Description			
This parameter defines the upper layer module to which the ClearTrcvWufFlagIndication from the Driver modules have to be routed. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00790]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserConfirmPnAvailabilityName		EcucFunctionNameDef	
BSW Description			
This parameter defines the name of <User_ConfirmPnAvailability>. If CanIfDispatchUserConfirmPnAvailabilityUL equals CAN_SM the name of <User_ConfirmPnAvailability> is fixed. If it equals CDD, the name is selectable. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00819]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserConfirmPnAvailabilityUL		EcucEnumerationParamDef	
BSW Description			
This parameter defines the upper layer module to which the ConfirmPnAvailability notification from the Driver modules have to be routed. If CanIfPublicPnSupport equals False, this parameter shall not be configurable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00820]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type	
CanIfDispatchUserCtrlBusOffName		EcucFunctionNameDef	
BSW Description			
This parameter defines the name of <User_ControllerBusOff>. This parameter depends on the parameter CanIfDispatchUserCtrlBusOffUL. If CanIfDispatchUserCtrlBusOffUL equals CAN_SM the name of <User_ControllerBusOff> is fixed. If CanIfDispatchUserCtrlBusOffUL equals CDD, the name of <User_ControllerBusOff> is selectable.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00525]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserCtrlBusOffUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the notifications of all ControllerBusOff events from the CAN Driver modules have to be routed via <User_ControllerBusOff>. There is no possibility to configure no upper layer (UL) module as the provider of <User_ControllerBusOff>.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00547]

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserCtrlModeIndicationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of <User_ControllerModeIndication>. This parameter depends on the parameter CanIfDispatchUserCtrlModeIndicationUL. If CanIfDispatchUserCtrlModeIndicationUL equals CAN_SM the name of <User_ControllerModeIndication> is fixed. If CanIfDispatchUserCtrlModeIndicationUL equals CDD, the name of <User_ControllerModeIndication> is selectable.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00683]

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserCtrlModeIndicationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the notifications of all ControllerTransition events from the CAN Driver modules have to be routed via <User_ControllerModeIndication>.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00684]

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserTrcvModelIndicationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of <User_TrvcModelIndication>. This parameter depends on the parameter CanIfDispatchUserTrcvModelIndicationUL. If CanIfDispatchUserTrcvModelIndicationUL equals CAN_SM the name of <User_TrvcModelIndication> is fixed. If CanIfDispatchUserTrcvModelIndicationUL equals CDD, the name of <User_TrvcModelIndication> is selectable.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00685]

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserTrcvModelIndicationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the notifications of all TransceiverTransition events from the CAN Transceiver Driver modules have to be routed via <User_TrvcModelIndication>. If no UL module is configured, no upper layer callback function will be called.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00686]

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		BSW Type
CanIfDispatchUserValidateWakeupEventName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of <User_ValidateWakeupEvent>. This parameter depends on the parameter CanIfDispatchUserValidateWakeupEventUL. If CanIfDispatchUserValidateWakeupEventUL equals ECUM, the name of <User_ValidateWakeupEvent> is fixed. If CanIfDispatchUserValidateWakeupEventUL equals CDD, the name of <User_ValidateWakeupEvent> is selectable.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00531]

BSW Module	BSW Context
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	BSW Type
CanIfDispatchUserValidateWakeupEventUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the notifications about positive former requested wake up sources have to be routed via <User_ValidateWakeupEvent>.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00549]

BSW Module	BSW Context
CanIf	CanIf
BSW Parameter	BSW Type
CanIfInitCfg	EcucParamConfContainerDef
BSW Description	
This container contains the init parameters of the CAN Interface.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00247]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfBufferCfg	EcucParamConfContainerDef
BSW Description	
This container contains the Txbuffer configuration. Multiple buffers with different sizes could be configured. If CanIfBufferSize (ECUC_CanIf_00834) equals 0, the CanIf Tx L-PDU only refers via this CanIfBufferCfg the corresponding CanIfHthCfg.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00832]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfBufferCfg	
BSW Parameter		BSW Type
CanIfBufferHthRef		EcucReferenceDef
BSW Description		
Reference to HTH, that defines the hardware object or the pool of hardware objects configured for transmission. All the CanIf Tx L-PDUs refer via the CanIfBufferCfg and this parameter to the HTHs if TxBuffering is enabled, or not.		
Each HTH shall not be assigned to more than one buffer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00833]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfBufferCfg	
BSW Parameter		BSW Type
CanIfBufferSize		EcucIntegerParamDef
BSW Description		
This parameter defines the number of CanIf Tx L-PDUs which can be buffered in one Txbuffer. If this value equals 0, the CanIf does not perform Txbuffering for the CanIf Tx L-PDUs which are assigned to this Txbuffer. If CanIfPublicTxBuffering equals False, this parameter equals 0 for all TxBuffer. If the CanHandleType of the referred HTH equals FULL, this parameter equals 0 for this TxBuffer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00834]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		BSW Type
CanIfInitCfgSet		EcucStringParamDef
BSW Description		

Selects the CAN Interface specific configuration setup. This type of the external data structure shall contain the post build initialization data for the CAN Interface for all underlying CAN Drivers.	
constant to CanIf_ConfigType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00623]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfInitHohCfg	EcucParamConfContainerDef
BSW Description	
This container contains the references to the configuration setup of each underlying CAN Driver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00257]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg
BSW Parameter	BSW Type
CanIfHrhCfg	EcucParamConfContainerDef
BSW Description	
This container contains configuration parameters for each hardware receive object (HRH).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00259]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	BSW Type
CanIfHrhCanCtrlIdRef	EcucReferenceDef
BSW Description	
Reference to controller Id to which the HRH belongs to. A controller can contain one or more HRHs.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00631]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	BSW Type
CanIfHrhIdSymRef	EcucReferenceDef
BSW Description	
The parameter refers to a particular HRH object in the CanDrv configuration (see CanHardwareObject ECUC_Can_00324).	
CanIf receives the following information of the CanDrv module by this reference:	
- CanHandleType (see ECUC_Can_00323)	
- CanObjectId (see ECUC_Can_00326)	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00634]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	BSW Type
CanIfHrhRangeCfg	EcucParamConfContainerDef
BSW Description	
Defines the parameters required for configuring multiple CANID ranges for a given same HRH.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00628]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	BSW Type
CanIfHrhRangeBaseId	EcucIntegerParamDef
BSW Description	
CAN Identifier used as base value in combination with CanIfHrhRangeMask for a masked ID range in which all CAN Ids shall pass the software filtering. The size of this parameter is limited by CanIfHrhRangeRxPduRangeCanIdType.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00825]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	BSW Type
CanIfHrhRangeMask	EcucIntegerParamDef
BSW Description	
Used as mask value in combination with CanIfHrhRangeBaselId for a masked ID range in which all CAN Ids shall pass the software filtering. The size of this parameter is limited by CanIfHrhRangeRxPduRangeCanIdType.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00826]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	BSW Type
CanIfHrhRangeRxPduLowerCanId	EcucIntegerParamDef
BSW Description	
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids shall pass the software filtering.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00629]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	BSW Type
CanIfHrhRangeRxPduRangeCanIdType	EcucEnumerationParamDef
BSW Description	
Specifies whether a configured Range of CAN Ids shall only consider standard CAN Ids or extended CAN Ids.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00644]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	BSW Type
CanIfHrhRangeRxPduUpperCanId	EcucIntegerParamDef
BSW Description	
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids shall pass the software filtering.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00630]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	BSW Type
CanIfHrhSoftwareFilter	EcucBooleanParamDef
BSW Description	
Selects the hardware receive objects by using the HRH range/list from CAN Driver configuration to define, for which HRH a software filtering has to be performed at during receive processing.	
True: Software filtering is enabled False: Software filtering is enabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00632]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg
BSW Parameter	BSW Type
CanIfHthCfg	EcucParamConfContainerDef
BSW Description	
This container contains parameters related to each HTH.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00258]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg
BSW Parameter	BSW Type
CanIfHthCanCtrlIdRef	EcucReferenceDef
BSW Description	
Reference to controller Id to which the HTH belongs to. A controller can contain one or more HTHs.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00625]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg
BSW Parameter	BSW Type
CanIfHthIdSymRef	EcucReferenceDef
BSW Description	
The parameter refers to a particular HTH object in the CanDrv configuration (see CanHardwareObject ECUC_Can_00324).	
CanIf receives the following information of the CanDrv module by this reference:	
<ul style="list-style-type: none"> - CanHandleType (see ECUC_Can_00323) - CanObjectId (see ECUC_Can_00326) 	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00627]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfMaxBufferSize	EcucIntegerParamDef
BSW Description	
Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00828]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfMaxRxPduCfg	EcucIntegerParamDef
BSW Description	
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00830]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfMaxTxPduCfg	EcucIntegerParamDef
BSW Description	
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00829]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg
BSW Parameter	BSW Type
CanIfRxPduCfg	EcucParamConfContainerDef
BSW Description	

This container contains the configuration (parameters) of each receive CAN L-PDU.	
The SHORT-NAME of "CanIfRxPduConfig" container itself represents the symbolic name of Receive L-PDU.	
This L-SDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00249]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduCanId	EcucIntegerParamDef
BSW Description	
CAN Identifier of Receive CAN L-PDUs used by the CAN Interface. Exa: Software Filtering. This parameter is used if exactly one Can Identifier is assigned to the Pdu. If a range is assigned then the CanIfRxPduCanIdRange parameter shall be used.	
Range: 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier	
Template Description	
This attribute is used to define the identifier this frame shall use on the CAN network.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00598]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduCanIdMask	EcucIntegerParamDef
BSW Description	
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter defines a CAN Identifier range in an alternative way to CanIfRxPduCanIdRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier. Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.	
Template Description	
Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.rxMask	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00822]

BSW Module		BSW Context	
CanIf		CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		BSW Type	
CanIfRxPduCanIdRange		EcucParamConfContainerDef	
BSW Description			
Optional container that allows to map a range of CAN Ids to one PduId.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00743]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange	
BSW Parameter		BSW Type	
CanIfRxPduCanIdRangeLowerCanId		EcucIntegerParamDef	
BSW Description			
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00745]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange	
BSW Parameter		BSW Type	
CanIfRxPduCanIdRangeUpperCanId		EcucIntegerParamDef	
BSW Description			
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00744]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		BSW Type	

CanIfRxPduCanIdType	EcucEnumerationParamDef
BSW Description	
CAN Identifier of receive CAN L-PDUs used by the CAN Driver for CAN L-PDU reception.	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameRxBehavior: Defines which CAN protocol shall be expected for frame reception.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRxBehavior	
Mapping Rule	Mapping Type
Mapping fully defined by all permutations of canAddressingMode and canFrameRxBehavior.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00596]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
EXTENDED_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN 2.0 or CAN FD frame with extended identifier (29 bits)	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameRxBehavior: Defines which CAN protocol shall be expected for frame reception.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRxBehavior	
Mapping Rule	Mapping Type
canAddressingMode = "extended" and canFrameRxBehavior = "any".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
EXTENDED_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN FD frame with extended identifier (29 bits)	
Template Description	

CanFrameTriggering.canAddressingMode:	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
CanFrameTriggering.canFrameRxBehavior:	
Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "extended" and canFrameRxBehavior = "canFd".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
EXTENDED_NO_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN 2.0 frame with extended identifier (29 bits)	
Template Description	
CanFrameTriggering.canAddressingMode:	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
CanFrameTriggering.canFrameRxBehavior:	
Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "extended" and canFrameRxBehavior = "can20".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
STANDARD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN 2.0 or CAN FD frame with standard identifier (11 bits)	
Template Description	

CanFrameTriggering.canAddressingMode:	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
CanFrameTriggering.canFrameRxBehavior:	
Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "standard" and canFrameRxBehavior = "any".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
STANDARD_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN FD frame with standard identifier (11 bits)	
Template Description	
CanFrameTriggering.canAddressingMode:	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
CanFrameTriggering.canFrameRxBehavior:	
Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "standard" and canFrameRxBehavior = "canFd".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdType
BSW Parameter	BSW Type
STANDARD_NO_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN 2.0 frame with standard identifier (11 bits)	
Template Description	

CanFrameTriggering.canAddressingMode:	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
CanFrameTriggering.canFrameRxBehavior:	
Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameRxBehavior	
Mapping Rule	Mapping Type
canAddressingMode = "standard" and canFrameRxBehavior = "can20".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduDataLength	EcucIntegerParamDef
BSW Description	
Data length of the received CAN L-PDUs used by the CAN Interface. This information is used for Data Length Check. Additionally it might specify the valid bits in case of the discrete DLC for CAN FD L-PDUs > 8 bytes.	
The data area size of a CAN L-PDU can have a range from 0 to 64 bytes.	
Template Description	
The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay).	
The frameLength of zero bytes is allowed.	
Please consider also TPS_SYST_02255.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame.frameLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00599]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduDataLengthCheck	EcucBooleanParamDef
BSW Description	
This parameter switches the message specific data length check. True: Data length check will be executed during the reception of this PDU. False: No data length check will be executed during the reception of this PDU.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00846]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		BSW Type
CanIfRxPduHrhIdRef		EcucReferenceDef
BSW Description		
The HRH to which Rx L-PDU belongs to, is referred through this parameter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00602]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		BSW Type
CanIfRxPdulId		EcucIntegerParamDef
BSW Description		
ECU wide unique, symbolic handle for receive CAN L-SDU. It shall fulfill ANSI/AUTOSAR definitions for constant defines.		
Range: 0..max. number of defined CanRxPdulds		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00597]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		BSW Type
CanIfRxPduReadData		EcucBooleanParamDef
BSW Description		
Enables and disables the Rx buffering for reading of received L-SDU data.		
True: Enabled False: Disabled		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00600]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduReadNotifyStatus	EcucBooleanParamDef
BSW Description	
Enables and disables receive indication for each receive CAN L-SDU for reading its notification status.	
True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00595]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00601]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduUserRxIndicationName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter CanIfRxPduUserRxIndicationUL. If CanIfRxPduUserRxIndicationUL equals CAN_TP, CAN_NM, PDUR, XCP, CAN_TSYN, J1939NM or J1939TP, the name of the <User_RxIndication> is fixed. If CanIfRxPduUserRxIndicationUL equals CDD, the name of the <User_RxIndication> is selectable.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00530]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfRxPduUserRxIndicationUL	EcucEnumerationParamDef
BSW Description	
<p>This parameter defines the upper layer (UL) module to which the indication of the successfully received CANRXPDUID has to be routed via <User_RxIndication>. This <User_RxIndication> has to be invoked when the indication of the configured CANRXPDUID will be received by an Rx indication event from the CAN Driver module. If no upper layer (UL) module is configured, no <User_RxIndication> has to be called in case of an Rx indication event of the CANRXPDUID from the CAN Driver module.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00529]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	BSW Type
CanIfTTRxFrameTriggering	EcucParamConfContainerDef
BSW Description	
<p>CanIfTTRxFrameTriggering is specified in the SWS TTCAN Interface and defines Frame trigger for TTCAN reception.</p> <p>This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and a joblist is used for reception.</p>	
Template Description	
CAN specific attributes to the FrameTriggering	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00003]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfTTRxFrameTriggering

BSW Parameter		BSW Type
CanIfTTRxHwObjectTriggerIdRef		EcucReferenceDef
BSW Description		
This parameter refers to a particular TTCAN hardware receive object Trigger of a hardware object in the TTCAN Driver Module, which is referred via plain CAN parameter CANIF_HRH_HANDLETYPE_REF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00133]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfTTRxFrameTriggering	
BSW Parameter		BSW Type
CanTTRxJoblistTimeMark		EcucIntegerParamDef
BSW Description		
Defines the point in time, when the joblist execution function (JLEF) shall be called for the referenced rx trigger. Value is given in cycle time. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00136]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		BSW Type
CanIfTxPduCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of a transmit CAN L-PDU. It has to be configured as often as a transmit CAN L-PDU is needed.		
The SHORT-NAME of "CanIfTxPduConfig" container represents the symbolic name of Transmit L-PDU.		
This L-SDU consumes a meta data item of type CAN_ID_32.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID

valid	[ECUC_CanIf_00248]
-------	--------------------

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTTTxFrameTriggering	EcucParamConfContainerDef	
BSW Description		
CanIfTTTxFrameTriggering is specified in the SWS TTCAN Interface and defines Frame trigger for TTCAN transmission.		
This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and a joblist is used.		
Template Description		
CAN specific attributes to the FrameTriggering		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00142]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTTTxFrameTriggering	
BSW Parameter	BSW Type	
CanIfTTTxHwObjectTriggerIdRef	EcucReferenceDef	
BSW Description		
This parameter refers to a particular TTCAN hardware transmit object Trigger of a hardware object in the TTCAN Driver Module, which is referred via plain CAN parameter CANIF_HTH_HANDLETYPE_REF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00128]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTTTxFrameTriggering	
BSW Parameter	BSW Type	
CanIfTTTxJoblistTimeMark	EcucIntegerParamDef	
BSW Description		
Defines the point in time, when the joblist execution function (JLEF) shall be called for the referenced tx frame trigger.		
Value is given in cycle time. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00132]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		BSW Type
CanIfTxPduBufferRef		EcucReferenceDef
BSW Description		
Configurable reference to a CanIf buffer configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00831]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		BSW Type
CanIfTxPduCanId		EcucIntegerParamDef
BSW Description		
CAN Identifier of transmit CAN L-PDUs used by the CAN Driver for CAN L-PDU transmission. Range: 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier		
The CAN Identifier may be omitted for dynamic transmit L-PDUs.		
Template Description		
This attribute is used to define the identifier this frame shall use on the CAN network.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00592]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		BSW Type
CanIfTxPduCanIdMask		EcucIntegerParamDef
BSW Description		
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter may be used to keep parts of the CAN Identifier of dynamic transmit L-PDUs static. Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.		
Template Description		
Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.		
M2 Parameter		

SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.txMask	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Canlf_00823]

BSW Module	BSW Context
Canlf	Canlf/CanlfInitCfg/CanlfTxPduCfg
BSW Parameter	BSW Type
CanlfTxPduCanIdType	EcucEnumerationParamDef
BSW Description	
Type of CAN Identifier of the transmit CAN L-PDU used by the CAN Driver module for CAN L-PDU transmission.	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameTxBehavior: Defines which CAN protocol shall be used for frame transmission.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameTx Behavior	
Mapping Rule	Mapping Type
Mapping fully defined by all permutations of canAddressingMode and canFrameTxBehavior.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Canlf_00590]

BSW Module	BSW Context
Canlf	Canlf/CanlfInitCfg/CanlfTxPduCfg/CanlfTxPduCanIdType
BSW Parameter	BSW Type
EXTENDED_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN frame with extended identifier (29 bits)	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameTxBehavior: Defines which CAN protocol shall be used for frame transmission.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameTx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "extended" and canFrameRxBehavior = "can20".	full
Mapping Status	ECUC Parameter ID

valid	
-------	--

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTxPduCanIdType
BSW Parameter	BSW Type
EXTENDED_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN FD frame with extended identifier (29 bits)	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameTxBehavior: Defines which CAN protocol shall be used for frame transmission.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameTx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "extended" and canFrameRxBehavior = "canFd".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTxPduCanIdType
BSW Parameter	BSW Type
STANDARD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN frame with standard identifier (11 bits)	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameTxBehavior: Defines which CAN protocol shall be used for frame transmission.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressing Mode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameTx Behavior	
Mapping Rule	Mapping Type
canAddressingMode = "standard" and canFrameRxBehavior = "can20".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTxPduCanIdType
BSW Parameter	BSW Type

STANDARD_FD_CAN	EcucEnumerationLiteralDef
BSW Description	
CAN FD frame with standard identifier (11 bits)	
Template Description	
<p>CanFrameTriggering.canAddressingMode: The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.</p> <p>CanFrameTriggering.canFrameTxBehavior: Defines which CAN protocol shall be used for frame transmission.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canFrameTxBehavior	
Mapping Rule	Mapping Type
canAddressingMode = "standard" and canFrameRxBehavior = "canFd".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg
BSW Parameter	BSW Type
CanIfTxPdulId	EcucIntegerParamDef
BSW Description	
ECU wide unique, symbolic handle for transmit CAN L-SDU.	
Range: 0..max. number of CanIfTxPdulIds	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00591]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg
BSW Parameter	BSW Type
CanIfTxPduPnFilterPdu	EcucBooleanParamDef
BSW Description	
If CanIfPublicPnFilterSupport is enabled, by this parameter PDUs could be configured which will pass the CanIfPnFilter. If there is no CanIfTxPduPnFilterPdu configured per controller, the corresponding controller applies no CanIfPnFilter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00773]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduReadNotifyStatus	EcucBooleanParamDef	
BSW Description		
Enables and disables transmit confirmation for each transmit CAN L-SDU for reading its notification status.		
True: Enabled False: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00589]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduRef	EcucReferenceDef	
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00603]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduTriggerTransmit	EcucBooleanParamDef	
BSW Description		
Determines if or if not CanIf shall use the trigger transmit API for this PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_CanIf_00840]
-------	--------------------

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduTruncation	EcucBooleanParamDef	
BSW Description		
Enables/disables truncation of PDUs that exceed the configured size.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00845]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduType	EcucEnumerationParamDef	
BSW Description		
Defines the type of each transmit CAN L-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00593]

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter	BSW Type	
CanIfTxPduUserTriggerTransmitName	EcucFunctionNameDef	
BSW Description		
<p>This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter CanIfTxPduUserTxConfirmationUL. If CanIfTxPduUserTxConfirmationUL equals CAN_TP, CAN_NM, PDUR, XCP, CAN_TSYN, J1939NM or J1939TP, the name of the <User_TriggerTransmit> is fixed. If CanIfTxPduUserTxConfirmationUL equals CDD, the name of the <User_TxConfirmation> is selectable.</p> <p>Please be aware that this parameter depends on the same parameter as CanIfTxPduUserTxConfirmationName. It shall be clear which upper layer is responsible for that PDU.</p>		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00842]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		BSW Type
CanIfTxPduUserTxConfirmationName		EcucFunctionNameDef
BSW Description		
<p>This parameter defines the name of the <User_TxConfirmation>.</p> <p>This parameter depends on the parameter CanIfTxPduUserTxConfirmationUL.</p> <p>If CanIfTxPduUserTxConfirmationUL equals CAN_TP, CAN_NM, PDUR, XCP, CAN_TSYN, J1939NM or J1939TP, the name of the <User_TxConfirmation> is fixed. If CanIfTxPduUserTxConfirmationUL equals CDD, the name of the <User_TxConfirmation> is selectable.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00528]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		BSW Type
CanIfTxPduUserTxConfirmationUL		EcucEnumerationParamDef
BSW Description		
<p>This parameter defines the upper layer (UL) module to which the confirmation of the successfully transmitted CanTxPduId has to be routed via the <User_TxConfirmation>.</p> <p>This <User_TxConfirmation> has to be invoked when the confirmation of the configured CanTxPduId will be received by a Tx confirmation event from the CAN Driver module.</p> <p>If no upper layer (UL) module is configured, no <User_TxConfirmation> has to be called in case of a Tx confirmation event of the CanTxPduId from the CAN Driver module.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00527]	

BSW Module	BSW Context	
CanIf	CanIf	
BSW Parameter		BSW Type
CanIfPrivateCfg		EcucParamConfContainerDef
BSW Description		
This container contains the private configuration (parameters) of the CAN Interface.		

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00245]

BSW Module	BSW Context
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	BSW Type
CanIfFixedBuffer	EcucBooleanParamDef
BSW Description	
<p>This parameter defines if the buffer element length shall be fixed to 8 Bytes for buffers to which only PDUs < 8 Bytes are assigned.</p> <p>TRUE: Minimum buffer element length is fixed to 8 Bytes. FALSE: Buffer element length depends on the size of the referencing PDUs.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00827]

BSW Module	BSW Context
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	BSW Type
CanIfPrivateDataLengthCheck	EcucBooleanParamDef
BSW Description	
<p>Selects whether Data Length Check is supported.</p> <p>True: Enabled False: Disabled</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00617]

BSW Module	BSW Context
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	BSW Type
CanIfPrivateSoftwareFilterType	EcucEnumerationParamDef

BSW Description	
<p>Selects the desired software filter mechanism for reception only. Each implemented software filtering method is identified by this enumeration number.</p> <p>Range: Types implemented software filtering methods</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00619]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPrivateCfg	
BSW Parameter		BSW Type
CanIfSupportTTCAN		EcucBooleanParamDef
BSW Description		
<p>Defines whether TTCAN is supported.</p> <p>TRUE: TTCAN is supported. FALSE: TTCAN is not supported, only normal CAN communication is possible.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00675]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPrivateCfg	
BSW Parameter		BSW Type
CanIfTTGeneral		EcucParamConfContainerDef
BSW Description		
<p>CanIfTTGeneral is specified in the SWS TTCAN Interface and defines if and in which way TTCAN is supported.</p> <p>This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and used.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00005]

BSW Module		BSW Context	
CanIf		CanIf/CanIfPrivateCfg/CanIfTTGeneral	
BSW Parameter		BSW Type	
CanIfTTDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00835]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfPrivateCfg/CanIfTTGeneral/CanIfTTDemEventParameterRefs	
BSW Parameter		BSW Type	
CANIF_TT_E_JLE_SYNC		EcucReferenceDef	
BSW Description			
Reference to configured DEM event to report that the JLEF lost synchronization to the local time of the TTCAN controller.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00836]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfPrivateCfg/CanIfTTGeneral	
BSW Parameter		BSW Type	
CanIfTTJoblist		EcucBooleanParamDef	
BSW Description			
Defines whether TTCAN is processed via a joblist. TRUE: Joblist is used. FALSE: No joblist is used.			
This parameter is only configurable if TTCAN is enabled by parameter CanIfSupportTTCAN.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00126]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfPrivateCfg/CanIfTTGeneral	
BSW Parameter		BSW Type	
CanIfTTMaxIsrDelay		EcucIntegerParamDef	
BSW Description			
Defines the maximum delay for the execution of the joblist execution function JLEF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00127]	

BSW Module		BSW Context	
CanIf		CanIf	
BSW Parameter		BSW Type	
CanIfPublicCfg		EcucParamConfContainerDef	
BSW Description			
This container contains the public configuration (parameters) of the CAN Interface.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00246]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfPublicCfg	
BSW Parameter		BSW Type	
CanIfBusMirroringSupport		EcucBooleanParamDef	
BSW Description			
Enable support for Bus Mirroring.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanIf_00847]	

BSW Module		BSW Context	
CanIf		CanIf/CanIfPublicCfg	
BSW Parameter		BSW Type	
CanIfDevErrorDetect		EcucBooleanParamDef	

BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00614]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfMetaDataSupport	EcucBooleanParamDef
BSW Description	
Enable support for dynamic ID handling using L-SDU MetaData.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00824]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicCddHeaderFile	EcucStringParamDef
BSW Description	
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00671]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicHandleTypeEnum	EcucEnumerationParamDef
BSW Description	

This parameter is used to configure the Can_HwHandleType. The Can_HwHandleType represents the hardware object handles of a CAN hardware unit. For CAN hardware units with more than 255 HW objects the extended range shall be used (UINT16).

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00742]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicIcomSupport	EcucBooleanParamDef
BSW Description	
Selects support of Pretended Network features in CanIf. True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00839]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicMultipleDrvSupport	EcucBooleanParamDef
BSW Description	
Selects support for multiple CAN Drivers. True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00612]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type

CanIfPublicPnSupport	EcucBooleanParamDef
BSW Description	
Selects support of Partial Network features in CanIf. True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00772]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicReadRxPduDataApi	EcucBooleanParamDef
BSW Description	
Enables / Disables the API CanIf_ReadRxPduData() for reading received L-SDU data. True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00607]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfPublicReadRxPduNotifyStatusApi	EcucBooleanParamDef
BSW Description	
Enables and disables the API for reading the notification status of receive L-PDUs. True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00608]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		BSW Type
CanIfPublicReadTxPduNotifyStatusApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reading the notification status of transmit L-PDUs.		
True: Enabled False: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00609]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		BSW Type
CanIfPublicSetDynamicTxIdApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reconfiguration of the CAN Identifier for each Transmit L-PDU.		
True: Enabled False: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanIf_00610]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		BSW Type
CanIfPublicTxBuffering		EcucBooleanParamDef
BSW Description		
Enables and disables the buffering of transmit L-PDUs (rejected by the CanDrv) within the CAN Interface module.		
True: Enabled False: Disabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00618]

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter	BSW Type	
CanIfPublicTxConfirmPollingSupport	EcucBooleanParamDef	
BSW Description		
Configuration parameter to enable/disable the API to poll for Tx Confirmation state.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00733]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter	BSW Type	
CanIfPublicWakeupCheckValidByNM	EcucBooleanParamDef	
BSW Description		
<p>If enabled, only NM messages shall validate a detected wake-up event in CanIf. If disabled, all received messages corresponding to a configured Rx PDU shall validate such a wake-up event. This parameter depends on CanIfPublicWakeupCheckValidSupport and shall only be configurable, if it is enabled.</p> <p>True: Enabled False: Disabled</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanIf_00741]	

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter	BSW Type	
CanIfPublicWakeupCheckValidSupport	EcucBooleanParamDef	
BSW Description		
Selects support for wake up validation		
<p>True: Enabled False: Disabled</p>		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00611]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfSetBaudrateApi	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable the CanIf_SetBaudrate API to change the baud rate of a CAN Controller. If this parameter is set to true the CanIf_SetBaudrate API shall be supported. Otherwise the API is not supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00838]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfTriggerTransmitSupport	EcucBooleanParamDef
BSW Description	
Enables the CanIf_TriggerTransmit API at Pre-Compile-Time. Therefore, this parameter defines if there shall be support for trigger transmit transmissions. TRUE: Enabled FALSE: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00844]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfTxOfflineActiveSupport	EcucBooleanParamDef
BSW Description	
Determines whether TxOffLineActive feature (see SWS_CANIF_00072) is supported by CanIf. True: Enabled False: Disabled	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00837]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables and disables the API for reading the version information about the CAN Interface.	
True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00613]

BSW Module	BSW Context
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	BSW Type
CanIfWakeupSupport	EcucBooleanParamDef
BSW Description	
Enables the CanIf_CheckWakeup API at Pre-Compile-Time. Therefore, this parameter defines if there shall be support for wake-up. TRUE: Enabled FALSE: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00843]

BSW Module	BSW Context
CanIf	CanIf
BSW Parameter	BSW Type
CanIfTrcvDrvCfg	EcucParamConfContainerDef
BSW Description	

This container contains the configuration (parameters) of all addressed CAN transceivers by each underlying CAN Transceiver Driver module. For each CAN transceiver Driver a separate instance of this container shall be provided.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00273]

BSW Module	BSW Context
CanIf	CanIf/CanIfTrcvDrvCfg
BSW Parameter	BSW Type
CanIfTrcvCfg	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of one addressed CAN transceiver by the underlying CAN Transceiver Driver module. For each CAN transceiver a separate instance of this container has to be provided.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00587]

BSW Module	BSW Context
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg
BSW Parameter	BSW Type
CanIfTrcvCanTrcvRef	EcucReferenceDef
BSW Description	
This parameter references to the logical handle of the underlying CAN transceiver from the CAN transceiver driver module to be served by the CAN Interface module.	
Range: 0..max. number of underlying supported CAN transceivers	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00605]

BSW Module	BSW Context
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg
BSW Parameter	BSW Type

CanIfTrcvId	EcucIntegerParamDef
BSW Description	
This parameter abstracts from the CAN Transceiver Driver specific parameter Transceiver. Each transceiver of all connected CAN Transceiver Driver modules shall be assigned to one specific TransceiverId of the CanIf.	
Range: 0..number of configured transceivers of all CAN Transceiver Driver modules	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00654]

BSW Module	BSW Context
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg
BSW Parameter	BSW Type
CanIfTrcvWakeupSupport	EcucBooleanParamDef
BSW Description	
This parameter defines if a respective transceiver of the referenced CAN Transceiver Driver modules is queriable for wake up events.	
True: Enabled False: Disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanIf_00606]

C.2.3 Can Transceiver Mapping

BSW Module	BSW Context
CanTrcv	CanTrcv
BSW Parameter	BSW Type
CanTrcvConfigSet	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR WdgM module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID

valid	[ECUC_CanTrcv_00173]
-------	----------------------

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet	
BSW Parameter	BSW Type	
CanTrcvChannel	EcucParamConfContainerDef	
BSW Description		
Container gives CAN transceiver driver information about a single CAN transceiver (channel).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00143]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter	BSW Type	
CanTrcvAccess	EcucChoiceContainerDef	
BSW Description		
Container gives CanTrcv Driver information about access to a single CAN transceiver.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00101]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess	
BSW Parameter	BSW Type	
CanTrcvDioAccess	EcucParamConfContainerDef	
BSW Description		
Container gives CAN transceiver driver information about accessing ports and port pins. In addition relation between CAN transceiver hardware pin names and Dio port access information is given. If a CAN transceiver hardware has no Dio interface, there is no instance of this container.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00145]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess	
BSW Parameter	BSW Type	
CanTrcvDioChannelAccess	EcucParamConfContainerDef	
BSW Description		
Container gives DIO channel access by single Can transceiver channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00157]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess	
BSW Parameter	BSW Type	
CanTrcvDioSymNameRef	EcucChoiceReferenceDef	
BSW Description		
Choice Reference to a DIO Port, DIO Channel or DIO Channel Group. This reference replaces the CANTRCV_DIO_PORT_SYM_NAME, CANTRCV_DIO_CHANNEL_SYM_NAME and CANTRCV_DIO_GROUP_SYM_NAME references in the Can Trcv SWS.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00149]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess	
BSW Parameter	BSW Type	
CanTrcvHardwareInterfaceName	EcucStringParamDef	
BSW Description		
CAN transceiver hardware interface name. It is typically the name of a pin. From a Dio point of view it is either a port, a single channel or a channel group. Depending on this fact either CANTRCV_DIO_PORT_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_GROUP_SYMBOLIC_NAME shall reference a Dio configuration. The CAN transceiver driver implementation description shall list up this name for the appropriate CAN transceiver hardware.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00150]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess	
BSW Parameter	BSW Type	
CanTrcvSpiAccess	EcucParamConfContainerDef	
BSW Description		
Container gives CAN transceiver driver information about accessing Spi. If a CAN transceiver hardware has no Spi interface, there is no instance of this container.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTrcv_00183]	

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess	
BSW Parameter	BSW Type	
CanTrcvSpiSequence	EcucParamConfContainerDef	
BSW Description		
Container gives CAN transceiver driver information about one SPI sequence. One SPI sequence used by CAN transceiver driver is in exclusive use for it. No other driver is allowed to access this sequence. CAN transceiver driver may use one sequence to access n CAN transceiver hardware chips of the same type or n sequences are used to access one single CAN transceiver hardware chip. If a CAN transceiver hardware has no SPI interface, there is no instance of this container.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTrcv_00144]	

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence	
BSW Parameter	BSW Type	
CanTrcvSpiAccessSynchronous	EcucBooleanParamDef	
BSW Description		

This parameter is used to define whether the access to the Spi sequence is synchronous or asynchronous.	
true: SPI access is synchronous. false: SPI access is asynchronous.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00176]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence
BSW Parameter	BSW Type
CanTrcvSpiSequenceName	EcucReferenceDef
BSW Description	
Reference to a Spi sequence configuration container.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00151]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvChannelEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the CAN transceiver channel to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the CAN transceiver driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00194]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvChannelId	EcucIntegerParamDef

BSW Description	
Unique identifier of the CAN Transceiver Channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00155]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvChannelUsed	EcucBooleanParamDef
BSW Description	
Shall the related CAN transceiver channel be used?	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00096]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvControlsPowerSupply	EcucBooleanParamDef
BSW Description	
Is ECU power supply controlled by this transceiver? TRUE = Controlled by transceiver. FALSE = Not controlled by transceiver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00097]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00188]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvDemEventParameterRefs
BSW Parameter	BSW Type
CANTRCV_E_BUS_ERROR	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when bus error has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00189]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvHwPnSupport	EcucBooleanParamDef
BSW Description	
Indicates whether the HW supports the selective wake-up function	
TRUE = Selective wakeup feature is supported by the transceiver	
FALSE = Selective wakeup functionality is not available in transceiver	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00160]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvIcuChannelRef	EcucReferenceDef

BSW Description	
Reference to the IcuChannel to enable/disable the interrupts for wakeups.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00185]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvInitState	EcucEnumerationParamDef
BSW Description	
State of CAN transceiver after call to CanTrcv_Init.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00146]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvMaxBaudrate	EcucIntegerParamDef
BSW Description	
Indicates the data transfer rate in kbps. Maximum data transfer rate in kbps for transceiver hardware type. Only used for validation purposes. This value can be used by configuration tools.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00147]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvPartialNetwork	EcucParamConfContainerDef
BSW Description	

Container gives CAN transceiver driver information about the configuration of Partial Networking functionality.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00161]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	BSW Type
CanTrcvBaudRate	EcucIntegerParamDef
BSW Description	
Indicates the data transfer rate in kbps.	
Template Description	
Channels speed in bits/s.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate	
Mapping Rule	Mapping Type
CanTrcvBaudRate = SystemTemplate baudrate is in bps, so divide it by 1000 to get kbps	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00169]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	BSW Type
CanTrcvBusErrFlag	EcucBooleanParamDef
BSW Description	
Indicates if the Bus Error (BUSERR) flag is managed by the BSW. This flag is set if a bus failure is detected by the transceiver. TRUE = Supported by transceiver and managed by BSW. FALSE = Not managed by BSW.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00171]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	BSW Type
CanTrcvPnCanIdsExtended	EcucBooleanParamDef
BSW Description	

Indicates whether extended or standard ID is used. TRUE = Extended Can identifier is used. FALSE = Standard Can identifier is used	
Template Description	
Defines whether pncWakeupCanId and pncWakeupCanIdMask shall be interpreted as extended or standard CAN ID.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanIdExtended	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00164]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	BSW Type
CanTrcvPnEnabled	EcucBooleanParamDef
BSW Description	
Indicates whether the selective wake-up function is enabled or disabled in HW. TRUE = Selective wakeup feature is enabled in the transceiver hardware FALSE = Selective wakeup feature is disabled in the transceiver hardware	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00172]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	BSW Type
CanTrcvPnFrameCanId	EcucIntegerParamDef
BSW Description	
CAN ID of the Wake-up Frame (WUF).	
Template Description	
CAN Identifier used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00163]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork

BSW Parameter		BSW Type
CanTrcvPnFrameCanIdMask		EcucIntegerParamDef
BSW Description		
ID Mask for the selective activation of the transceiver. It is used to enable Frame Wake-up (WUF) on a group of IDs.		
Template Description		
Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanIdMask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00162]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		BSW Type
CanTrcvPnFrameDataMaskSpec		EcucParamConfContainerDef
BSW Description		
Defines data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00165]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec	
BSW Parameter		BSW Type
CanTrcvPnFrameDataMask		EcucIntegerParamDef
BSW Description		
Defines the n byte (Byte0 = LSB) of the data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00166]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec	
BSW Parameter		BSW Type
CanTrcvPnFrameDataMaskIndex		EcucIntegerParamDef
BSW Description		
holds the position n in frame of the mask-part		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00167]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		BSW Type
CanTrcvPnFrameDlc		EcucIntegerParamDef
BSW Description		
Data Length of the Wake-up Frame (WUF).		
Template Description		
Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDlc		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00168]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		BSW Type
CanTrcvPowerOnFlag		EcucBooleanParamDef
BSW Description		
Description: Indicates if the Power On Reset (POR) flag is available and is managed by the transceiver.		
TRUE = Supported by Hardware. FALSE = Not supported by Hardware		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00170]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter	BSW Type	
CanTrcvPorWakeupSourceRef	EcucReferenceDef	
BSW Description		
Symbolic name reference to specify the wakeup sources that should be used in the calls to EcuM_SetWakeupEvent as specified in [SWS_CanTrcv_00183] and [SWS_CanTrcv_00184]. This reference is mandatory if the HW supports POR or SYSERR flags		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00181]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter	BSW Type	
CanTrcvSyserrWakeupSourceRef	EcucReferenceDef	
BSW Description		
Symbolic name reference to specify the wakeup sources that should be used in the calls to EcuM_SetWakeupEvent as specified in [SWS_CanTrcv_00183] and [SWS_CanTrcv_00184]. This reference is mandatory if the HW supports POR or SYSERR flags		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00182]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter	BSW Type	
CanTrcvWakeupByBusUsed	EcucBooleanParamDef	
BSW Description		
Is wake up by bus supported? If CAN transceiver hardware does not support wake up by bus value is always FALSE. If CAN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not. TRUE = Is used. FALSE = Is not used.		
Template Description		

Defines whether the ECU shall be woken up by this CommunicationController.	
TRUE: wake up is possible	
FALSE: wake up is not supported	
Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationController.wakeUpByControllerSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00148]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	BSW Type
CanTrcvWakeupSourceRef	EcucReferenceDef
BSW Description	
Reference to a wakeup source in the EcuM configuration.	
This reference is only needed if CanTrcvWakeupByBusUsed is true.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00177]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet
BSW Parameter	BSW Type
CanTrcvSPICommRetries	EcucIntegerParamDef
BSW Description	
Indicates the maximum number of communication retries in case of a failed SPI communication (applies both to timed out communication and to errors/NACK in the response data). If configured value is '0', no retry is allowed (communication is expected to succeed at first try).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00175]

BSW Module	BSW Context
-------------------	--------------------

CanTrcv	CanTrcv/CanTrcvConfigSet	
BSW Parameter		BSW Type
CanTrcvSPICommTimeout		EcucIntegerParamDef
BSW Description		
Indicates the maximum time allowed to the CanTrcv for replying (either positively or negatively) to a SPI command. Timeout is configured in milliseconds. Timeout value of '0' means that no specific timeout is to be used by CanTrcv and the communication is executed at the best of the SPI HW capacity.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00174]

BSW Module	BSW Context	
CanTrcv	CanTrcv	
BSW Parameter		BSW Type
CanTrcvGeneral		EcucParamConfContainerDef
BSW Description		
Container gives CAN transceiver driver basic information.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00090]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		BSW Type
CanTrcvDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00152]

BSW Module	BSW Context
-------------------	--------------------

CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		BSW Type
CanTrcvEcucPartitionRef		EcucReferenceDef
BSW Description		
Maps the CAN transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The module will operate as an independent instance in each of the partitions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00193]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		BSW Type
CanTrcvIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00184]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		BSW Type
CanTrcvMainFunctionDiagnosticsPeriod		EcucFloatParamDef
BSW Description		
This parameter describes the period for cyclic call to CanTrcv_MainFunctionDiagnostics. Unit is seconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTrcv_00187]

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		BSW Type

CanTrcvMainFunctionPeriod	EcucFloatParamDef
BSW Description	
This parameter describes the period for cyclic call to CanTrcv_MainFunction. Unit is seconds.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_CanTrcv_00186]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvGeneral
BSW Parameter	BSW Type
CanTrcvTimerType	EcucEnumerationParamDef
BSW Description	
Type of the Time Service Predefined Timer.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_CanTrcv_00190]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvGeneral
BSW Parameter	BSW Type
CanTrcvVersionInfoApi	EcucBooleanParamDef
BSW Description	
Switches version information API on and off. If switched off, function need not be present in compiled code.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_CanTrcv_00153]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvGeneral
BSW Parameter	BSW Type
CanTrcvWaitTime	EcucFloatParamDef
BSW Description	
Wait time for transceiver state changes in seconds.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00191]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvGeneral
BSW Parameter	BSW Type
CanTrcvWakeUpSupport	EcucEnumerationParamDef
BSW Description	
Informs whether wake up is supported by polling or not supported. In case no wake up is supported by the hardware, setting has to be NOT_SUPPORTED. Only in the case of wake up supported by polling, function CanTrcv_MainFunction has to be present and to be invoked by the scheduler.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTrcv_00154]

C.2.4 CanNm Mapping

BSW Module	BSW Context
CanNm	CanNm
BSW Parameter	BSW Type
CanNmGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains the global configuration parameter of the CanNm. The parameters and the parameters of the sub containers shall be mapped to the C data type CanNm_ConfigType (for parameters where it is possible) which is passed to the CanNm_Init function.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00001]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmBusLoadReductionEnabled	EcucBooleanParamDef
BSW Description	

Pre-processor switch for enabling busload reduction support.	
Template Description	
Enables busload reduction support	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmClusterCoupling.nmBusloadReductionEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00040]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmBusSynchronizationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only.	
Template Description	
Enables bus synchronization support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00006]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmChannelConfig	EcucParamConfContainerDef
BSW Description	
This container contains the channel specific configuration parameter of the CanNm.	
Template Description	
Can specific NmCluster attributes	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster	
Mapping Rule	Mapping Type
Create container for each existing CanNmCluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00017]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmActiveWakeupBitEnabled	EcucBooleanParamDef
BSW Description	
Enables/Disables the handling of the Active Wakeup Bit in the CanNm module.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00084]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmAllNmMessagesKeepAwake		EcucBooleanParamDef
BSW Description		
Specifies if CanNm drops irrelevant NM PDUs.		
false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling		
true: Every NM PDU triggers the standard RX indication handling		
Template Description		
Specifies if Nm drops irrelevant NM PDUs.		
false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake		
true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmNode.allNmMessagesKeepAwake		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00068]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmBusLoadReductionActive		EcucBooleanParamDef
BSW Description		
This parameter defines if bus load reduction for the respective NM channel is active or not.		
Template Description		
It determines if bus load reduction for the respective CanNm channel is active or not.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmBusloadReductionActive		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00042]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmCarWakeUpBitPosition		EcucIntegerParamDef
BSW Description		

Specifies the Bit position of the CWU within the NM PDU.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	Mapping Type
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNmCarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00075]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmCarWakeUpBytePosition	EcucIntegerParamDef
BSW Description	
Specifies the Byte position of the CWU within the NM PDU.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	Mapping Type
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNmCarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NM PDU.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00076]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmCarWakeUpFilterEnabled	EcucBooleanParamDef
BSW Description	
If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNodeId is considered as CWU request. FALSE - CWU filtering is not supported TRUE - CWU filtering is supported	
Template Description	
If this attribute is set to true the CareWakeUp filtering is supported.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmNode.nmCarWakeUpFilterEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00077]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter		BSW Type	
CanNmCarWakeUpFilterNodeId		EcucIntegerParamDef	
BSW Description			
Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNodeId is considered as CWU request.			
Template Description			
Source node identifier for CarWakeUp filtering.			
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpFilterNodeId			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanNm_00078]	

BSW Module		BSW Context	
CanNm		CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type	
CanNmCarWakeUpRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables support of CarWakeUp bit evaluation in received NM PDUs. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported			
Template Description			
If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.			
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmNode.nmCarWakeUpRxEnabled			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanNm_00074]	

BSW Module		BSW Context	
CanNm		CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type	
CanNmComMNetworkHandleRef		EcucReferenceDef	
BSW Description			
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanNm_00018]	

BSW Module		BSW Context	
CanNm		CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type	

CanNmImmediateNmCycleTime	EcucFloatParamDef
BSW Description	
Defines the immediate NM PDU cycle time in seconds which is used for CanNmImmediateNmTransmissions NM PDU transmissions.	
Template Description	
Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This parameter is only valid if CanNmImmediateNmTransmissions is greater one.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmImmediateNmCycleTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00057]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmImmediateNmTransmissions	EcucIntegerParamDef
BSW Description	
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immediate NM PDUs is defined by CanNmImmediateNmCycleTime.	
Template Description	
Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmImmediateNmTransmissions	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00056]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmMsgCycleOffset	EcucFloatParamDef
BSW Description	
Time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
Template Description	
Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmNode.nmMsgCycleOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00029]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmMsgCycleTime	EcucFloatParamDef	
BSW Description		
Period of a NM PDU in seconds. It determines the periodic rate in the "periodic transmission mode with bus load reduction" and is the basis for transmit scheduling in the "periodic transmission mode without bus load reduction".		
Template Description		
Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmMsgCycleTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00028]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmMsgReducedTime	EcucFloatParamDef	
BSW Description		
Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.		
Template Description		
Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmNode.nmMsgReducedTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00043]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmMsgTimeoutTime	EcucFloatParamDef	
BSW Description		
When using Partial Network and this timeout is defined then CanNm monitors that a NM-PDU is transmitted successfully within this Transmission Timeout Time and provides an error notification otherwise.		
Template Description		
Timeout of an NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmMessageTimeoutTime		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00030]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
Precompile time switch to enable the node detection feature.		
Template Description		
Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNodeDetectionEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00088]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmNodeId		EcucIntegerParamDef
BSW Description		
Node identifier of local node.		
Template Description		
Node identifier of local NmNode. Shall be unique in the NmCluster.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmNodeId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00031]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmNodeIdEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the source node identifier.		
Template Description		
Enables the source node identifier.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNodeIdEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00090]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmPduCbvPosition	EcucEnumerationParamDef	
BSW Description		
Defines the position of the control bit vector within the NM PDU. The value of the parameter represents the location of the Control Bit Vector in the NM PDU (CanNmPduByte0 means byte 0, CanNmPduByte1 means byte 1, CanNmPduOff means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType		
Template Description		
Defines the position of the control bit vector within the NmPdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmCbvPosition		
Mapping Rule		Mapping Type
Derive byte position from nmCbvPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00026]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmPduNidPosition	EcucEnumerationParamDef	
BSW Description		
Defines the position of the source node identifier within the NM PDU. The value of the parameter represents the location of the source node identifier in the NM PDU (CANNM_PDU_BYTE_0 means byte 0, CANNM_PDU_BYTE_1 means byte 1, CANNM_PDU_OFF means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType		
Template Description		
Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmNidPosition		
Mapping Rule		Mapping Type
Derive byte position from nmNidPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00025]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter	BSW Type	
CanNmPnEnabled	EcucBooleanParamDef	
BSW Description		

Enables or disables support of partial networking.	
false: Partial networking Range not supported true: Partial networking supported	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmPncParticipation	
Mapping Rule	Mapping Type
If NmCluster.nmPncParticipation has the value "true" or is not defined then CanNmPnEnabled shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00066]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmPnEraCalcEnabled	EcucBooleanParamDef
BSW Description	
Specifies if CanNm calculates the PN request information for external requests. (ERA)	
false: PN request are not calculated true: PN request are calculated	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00067]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmPnEraRxNSduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack. The SduRef is required for every CanNm Channel, because ERA is reported per channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00079]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter		BSW Type
CanNmPnHandleMultipleNetworkRequests		EcucBooleanParamDef
BSW Description		
Specifies if CanNm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00073]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmRemoteSleepIndTime		EcucFloatParamDef
BSW Description		
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep.		
Template Description		
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmRemoteSleepIndicationTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00023]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmRepeatMessageTime		EcucFloatParamDef
BSW Description		
Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.		
Template Description		
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmRepeatMessageTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00022]

BSW Module	BSW Context	
------------	-------------	--

CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmRepeatMsgIndEnabled	EcucBooleanParamDef	
BSW Description		
Enable/disable the notification that a RepeatMessageRequest bit has been received.		
Template Description		
Switch for enabling the Repeat Message Bit Indication.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmRepeatMsgIndEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00089]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		BSW Type
CanNmRxPdu	EcucParamConfContainerDef	
BSW Description		
This container is used to configure the Rx PDU properties that are used for the CanNm Channel.		
Template Description		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.rxNmPdu		
Mapping Rule		Mapping Type
Create container for each NmPdu that is received on the regarded Nm cluster		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00038]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmRxPdu	
BSW Parameter		BSW Type
CanNmRxPduId	EcucIntegerParamDef	
BSW Description		
This parameter defines the Rx PDU ID of the CanIf L-PDU range that is associated with this CanNm channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00054]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmRxPdu	
BSW Parameter		BSW Type
CanNmRxPduRef	EcucReferenceDef	
BSW Description		

Reference to the global PDU that is used by this CanNm channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00039]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmStayInPbsEnabled	EcucBooleanParamDef
BSW Description	
If this parameter is disabled Prepare Bus-Sleep Mode is left after CanNmWaitBusSleepTime. If this parameter is enabled Prepare Bus-Sleep Mode can only be left if ECU is powered off or any restart reason applies.	
Template Description	
Switch for enabling the Repeat Message Bit Indication.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmRepeatMsgIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00092]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmTimeoutTime	EcucFloatParamDef
BSW Description	
Network Timeout for NM PDUs. It denotes the time in seconds how long the NM shall stay in the Ready Sleep State before transition into the Prepare Bus-Sleep Mode is initiated.	
Template Description	
Network Timeout for NmPdus in seconds It denotes the time how long the CanNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmNetworkTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00020]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmTxPdu	EcucParamConfContainerDef

BSW Description	
This container contains the CanNmTxConfirmationPduId and the CanNmTxPduRef.	
Template Description	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.txNmPdu	
Mapping Rule	Mapping Type
Create container for each NmPdu that is transmitted on the regarded Nmcluster	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00036]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmTxPdu
BSW Parameter	BSW Type
CanNmTxConfirmationPduId	EcucIntegerParamDef
BSW Description	
Handle Id to be used by the Lower Layer to confirm the transmission of the CanNmTxPdu to the LowerLayer.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00048]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmTxPdu
BSW Parameter	BSW Type
CanNmTxPduRef	EcucReferenceDef
BSW Description	
The reference to the common PDU structure.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00037]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmUserDataTxPdu	EcucParamConfContainerDef
BSW Description	
This optional container is used to configure the UserNm PDU. This container is only available if CanNmComUserDataSupport is enabled.	
Template Description	

An ISignalToIPduMapping describes the mapping of ISignals to ISignalPdus and defines the position of the ISignal within an ISignalPdu.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	Mapping Type
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00045]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu
BSW Parameter	BSW Type
CanNmTxUserDataPduId	EcucIntegerParamDef
BSW Description	
This parameter defines the Handle ID of the NM User Data I-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00047]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu
BSW Parameter	BSW Type
CanNmTxUserDataPduRef	EcucReferenceDef
BSW Description	
Reference to the NM User Data I-PDU in the global PDU collection.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00046]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	BSW Type
CanNmWaitBusSleepTime	EcucFloatParamDef
BSW Description	
Timeout for bus calm down phase. It denotes the time in seconds how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.	
Template Description	

Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.	
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmWaitBusSleepTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00021]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmComControlEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00013]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmComUserDataSupport	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping	
Mapping Rule	Mapping Type
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00044]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmCoordinatorSyncSupport	EcucBooleanParamDef
BSW Description	
Enables/disables the coordinator synchronization support.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00080]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00002]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmGlobalPnSupport	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling partial networking support globally.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00086]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmImmediateRestartEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the immediate transmission of a NM PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
Template Description	
Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
M2 Parameter	

SystemTemplate::NetworkManagement::CanNmClusterCoupling.nmImmediateRestartEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00009]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmImmediateTxconfEnabled	EcucBooleanParamDef
BSW Description	
Enable/disable the immediate tx confirmation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00041]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Call cycle in seconds of CanNm_MainFunction.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00032]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmPassiveModeEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling support of the Passive Mode.	
Template Description	
Enables support of the Passive Mode. The passive mode is configurable per channel.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmPassiveModeEnabled	
Mapping Rule	Mapping Type
1:1 mapping. nmNode.nmPassiveModeEnabled shall always have the same value in all Nm Clusters with the same bus protocol in the scope of one EcuInstance.	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00010]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter	BSW Type	
CanNmPduRxIndicationEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication.		
Template Description		
Switch for enabling the PDU Rx Indication.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00011]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter	BSW Type	
CanNmPnEiraCalcEnabled	EcucBooleanParamDef	
BSW Description		
Specifies if CanNm calculates the PN request information for internal an external requests. (EIRA) true: PN request are calculated false: PN request are not calculated		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00070]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter	BSW Type	
CanNmPnEiraRxNSduRef	EcucReferenceDef	
BSW Description		
Reference to a Pdu in the COM-Stack. Only one SduRef is required for CanNm because the EIRA is the aggregation over all Can Channels.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00072]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		BSW Type
CanNmPnInfo		EcucParamConfContainerDef
BSW Description		
PN information configuration		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00071]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo	
BSW Parameter		BSW Type
CanNmPnFilterMaskByte		EcucParamConfContainerDef
BSW Description		
PN information configuration		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		Mapping Type
For one EcuInstance all contributing CanCommunicationConnector.pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2^{63} this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanNm_00069]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo/CanNmPnFilterMaskByte	
BSW Parameter		BSW Type
CanNmPnFilterMaskByteIndex		EcucIntegerParamDef
BSW Description		
Index of the filter mask byte. Specifies the position within the filter mask byte array.		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		

Mapping Rule		Mapping Type
<p>For one EcuInstance all contributing CanCommunicationConnector.pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2^{63} this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00063]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo/CanNmPnFilterMaskByte	
BSW Parameter	BSW Type	
CanNmPnFilterMaskByteValue	EcuIntegerParamDef	
BSW Description		
Parameter to configure the filter mask byte.		
Template Description		
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		Mapping Type
<p>For one EcuInstance all contributing CanCommunicationConnector.pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2^{63} this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00064]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo	
BSW Parameter	BSW Type	
CanNmPnInfoLength	EcuIntegerParamDef	
BSW Description		
Specifies the length of the PN request information in the NM PDU.		
Template Description		
Length of the partial networking request release information vector (in bytes).		
M2 Parameter		
SystemTemplate::System.pncVectorLength		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00061]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo	
BSW Parameter	BSW Type	
CanNmPnInfoOffset	EcucIntegerParamDef	
BSW Description		
Specifies the offset of the PN request information in the NM PDU.		
Template Description		
Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.		
M2 Parameter		
SystemTemplate::System.pncVectorOffset		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00060]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter	BSW Type	
CanNmPnResetTime	EcucFloatParamDef	
BSW Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA. The value shall be the same for every channel. Thus it is a global config parameter.		
Template Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.pnResetTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanNm_00059]	

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter	BSW Type	
CanNmRemoteSleepIndEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only.		
Template Description		
Switch for enabling remote sleep indication support.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00055]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmStateChangeIndEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the CAN NM state change notification.	
Template Description	
Enables the CAN Network Management state change notification.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00012]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmUserDataEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling user data support.	
Template Description	
Switch for enabling user data support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00004]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	BSW Type
CanNmVersionInfoApi	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling version info API support.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00003]

C.2.5 CanTp Mapping

BSW Module		BSW Context	
CanTp		CanTp	
BSW Parameter		BSW Type	
CanTpConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters and sub containers of the AUTOSAR CanTp module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00290]	

BSW Module		BSW Context	
CanTp		CanTp/CanTpConfig	
BSW Parameter		BSW Type	
CanTpChannel		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters of the CanTp channel.			
Template Description			
Configuration parameters of the CanTp channel.			
M2 Parameter			
SystemTemplate::TransportProtocols::CanTpChannel			
Mapping Rule			Mapping Type
Create Container ifor each CanTpChannel that exist in ECU Extract.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00288]	

BSW Module		BSW Context	
CanTp		CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		BSW Type	
CanTpRxNSdu		EcucParamConfContainerDef	
BSW Description			
The following parameters needs to be configured for each CAN N-SDU that the CanTp module receives via the CanTpChannel. This N-SDU produces meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16 and ADDRESS_EXTENSION_8.			
Template Description			
M2 Parameter			
SystemTemplate::TransportProtocols::CanTpConnection.tpSdu			
Mapping Rule			Mapping Type
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is received.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00137]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpBs	EcucIntegerParamDef	
BSW Description		
Sets the number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification.		
Template Description		
The maximum number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification.		
Note: For reasons of buffer length, the CAN Transport Layer can adapt the BS value within the limit of this maximum BS		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.maxBlockSize		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00276]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpNAe	EcucParamConfContainerDef	
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.		
Template Description		
Declares which communication addressing mode is supported.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule	Mapping Type	
Create container if addressingFormat is set to "mixed".	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00284]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNAe	
BSW Parameter	BSW Type	
CanTpNAe	EcucIntegerParamDef	
BSW Description		
This parameter contains the transport protocol address extension value.		
Template Description		
If the mixed addressing format is used, this parameter contains the transport protocol address extension value.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddressExtensionValue		
Mapping Rule	Mapping Type	
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddressExtension.	full	

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00285]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpNSa	EcucParamConfContainerDef	
BSW Description		
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
Template Description		
Declares which communication addressing mode is supported.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule	Mapping Type	
Create container if addressingFormat is set to "extended".	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00253]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNSa	
BSW Parameter	BSW Type	
CanTpNSa	EcucIntegerParamDef	
BSW Description		
This parameter contains the transport protocol source address value.		
Template Description		
An ECUs TP address on the referenced channel. This represents the diagnostic Address.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress		
Mapping Rule	Mapping Type	
The CanTpConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00254]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpNTa	EcucParamConfContainerDef	
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
Template Description		
Declares which communication addressing mode is supported.		

M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat	
Mapping Rule	Mapping Type
Create container if addressingFormat is set to "extended".	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00139]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNTa
BSW Parameter	BSW Type
CanTpNTa	EcucIntegerParamDef
BSW Description	
This parameter contains the transport protocol target address value.	
Template Description	
An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress	
Mapping Rule	Mapping Type
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00255]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpNar	EcucFloatParamDef
BSW Description	
Value in seconds of the N_Ar timeout. N_Ar is the time for transmission of a CAN frame (any N_PDU) on the receiver side.	
Template Description	
This attribute states the timeout between the PDU transmit request of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the receiver side (for FC or AF). Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpNode.timeoutAr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00277]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpNbr	EcucFloatParamDef
BSW Description	
Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.	
Template Description	

Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.timeoutBr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00245]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpNcr	EcucFloatParamDef
BSW Description	
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.	
Template Description	
This parameter defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.timeoutCr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00279]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpRxAddressingFormat	EcucEnumerationParamDef
BSW Description	
Declares which communication addressing mode is supported for this RxNSdu. Definition of Enumeration values: CanTpStandard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00281]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type

CanTpRxNPdu	EcucParamConfContainerDef
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU consumes a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.dataPdu	
Mapping Rule	Mapping Type
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00256]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxNPdu
BSW Parameter	BSW Type
CanTpRxNPduld	EcucIntegerParamDef
BSW Description	
The N-PDU identifier attached to the RxNsdu is identified by CanTpRxNSduld.	
Each RxNsdu identifier is linked to only one SF/FF/CF N-PDU identifier. Nevertheless, in the case of extended or mixed addressing format, the same N-PDU identifier can be used for several N-SDU identifiers. The distinction is made by the N_TA or N_AE value (first data byte of SF or FF frames).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00258]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxNPdu
BSW Parameter	BSW Type
CanTpRxNPduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00257]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpRxNSduId	EcucIntegerParamDef	
BSW Description		
Unique identifier user by the upper layer to call CanTp_CancelReceive, CanTp_ChangeParameter and CanTp_ReadParameter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00301]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpRxNSduRef	EcucReferenceDef	
BSW Description		
Reference to a Pdu in the COM-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00241]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	BSW Type	
CanTpRxPaddingActivation	EcucEnumerationParamDef	
BSW Description		
Defines if the receive frame uses padding or not. This parameter is restricted to 8 byte N-PDUs.		
Definition of enumeration values:		
CanTpOn: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always ≥ 8 bytes in case of CAN 2.0)		
CanTpOff: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic - any valid DLC value).		
Note: The mandatory mapping to the next higher valid DLC value for N-PDUs with a length > 8 bytes is not affected by this parameter.		
Template Description		

This specifies wheter or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.	
true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes)	
false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.paddingActivation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00249]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpRxTaType	EcucEnumerationParamDef
BSW Description	
Declares the communication type of this Rx N-SDU.	
Template Description	
Network Target Address type.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.taType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00250]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxTaType
BSW Parameter	BSW Type
CANTP_FUNCTIONAL	EcucEnumerationLiteralDef
BSW Description	
Functional request type	
Template Description	
Functional request type	
M2 Parameter	
SystemTemplate::TransportProtocols::NetworkTargetAddressType.functional	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxTaType
BSW Parameter	BSW Type
CANTP_PHYSICAL	EcucEnumerationLiteralDef
BSW Description	

Physical request type	
Template Description	
Physical request type	
M2 Parameter	
SystemTemplate::TransportProtocols::NetworkTargetAddressType.physical	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpRxWftMax	EcucIntegerParamDef
BSW Description	
<p>This parameter indicates how many Flow Control wait N-PDUs can be consecutively transmitted by the receiver. It is local to the node and is not transmitted inside the FC protocol data unit.</p> <p>CanTpRxWftMax is used to avoid sender nodes being potentially hooked-up in case of a temporarily reception inability on the part of the receiver nodes, whereby the sender could be waiting continuously.</p>	
Template Description	
This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpNode.maxFcWait	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00251]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	BSW Type
CanTpSTmin	EcucFloatParamDef
BSW Description	
<p>Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.</p> <p>For further details on this parameter value see ISO 15765-2 specification.</p>	
Template Description	
Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpNode.stMin	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00252]

BSW Module	BSW Context
-------------------	--------------------

CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		BSW Type
CanTpTxFcNPdu	EcucParamConfContainerDef	
BSW Description		
Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU produces a meta data item of type CAN_ID_32.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.flowControlPdu		
Mapping Rule		Mapping Type
Create container if the CanTpConnection contains a reference to a FlowControl NPdu that is received by the regarded ECU.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00259]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpTxFcNPdu	
BSW Parameter		BSW Type
CanTpTxFcNPduConfirmationPduId	EcucIntegerParamDef	
BSW Description		
Handle Id to be used by the CanIf to confirm the transmission of the CanTpTxFcNPdu to the CanIf module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00287]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpTxFcNPdu	
BSW Parameter		BSW Type
CanTpTxFcNPduRef	EcucReferenceDef	
BSW Description		
Reference to a Pdu in the COM-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00260]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		BSW Type

CanTpTxNSdu	EcucParamConfContainerDef
BSW Description	
The following parameters needs to be configured for each CAN N-SDU that the CanTp module transmits via the CanTpChannel. This N-SDU consumes meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16 and ADDRESS_EXTENSION_8.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.tpSdu	
Mapping Rule	Mapping Type
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is transmitted.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00138]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	BSW Type
CanTpNAe	EcucParamConfContainerDef
BSW Description	
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat	
Mapping Rule	Mapping Type
Create container if addressingFormat is set to "mixed".	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00284]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNAe
BSW Parameter	BSW Type
CanTpNAe	EcucIntegerParamDef
BSW Description	
This parameter contains the transport protocol address extension value.	
Template Description	
If the mixed addressing format is used, this parameter contains the transport protocol address extension value.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpAddress.tpAddressExtensionValue	
Mapping Rule	Mapping Type
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddressExtension.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00285]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu

BSW Parameter		BSW Type
CanTpNSa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
Template Description		
Declares which communication addressing mode is supported.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		Mapping Type
Create container if addressingFormat is set to "extended".		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00253]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNSa	
BSW Parameter		BSW Type
CanTpNSa		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol source address value.		
Template Description		
An ECUs TP address on the referenced channel. This represents the diagnostic Address.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress		
Mapping Rule		Mapping Type
The CanTpConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00254]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpNTa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
Template Description		
Declares which communication addressing mode is supported.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		Mapping Type
Create container if addressingFormat is set to "extended".		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00139]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNTa	
BSW Parameter		BSW Type
CanTpNTa		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol target address value.		
Template Description		
An ECUs TP address on the referenced channel. This represents the diagnostic Address.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress		
Mapping Rule		Mapping Type
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00255]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpNas		EcucFloatParamDef
BSW Description		
Value in second of the N_As timeout. N_As is the time for transmission of a CAN frame (any N_PDU) on the part of the sender.		
Template Description		
This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpNode.timeoutAs		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00263]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpNbs		EcucFloatParamDef
BSW Description		
Value in seconds of the N_Bs timeout. N_Bs is the time of transmission until reception of the next Flow Control N_PDU.		
Template Description		
This parameter defines the timeout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.		
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.timeoutBs		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00264]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter	BSW Type	
CanTpNcs	EcucFloatParamDef	
BSW Description	Value in seconds of the performance requirements relating to N_Cs. CanTpNcs is the time in which CanTp is allowed to request from PduR the Tx data of a Consecutive Frame N_PDU.	
Template Description	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.	
M2 Parameter	SystemTemplate::TransportProtocols::CanTpConnection.timeoutCs	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00265]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter	BSW Type	
CanTpRxFcNPdu	EcucParamConfContainerDef	
BSW Description	Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU consumes a meta data item of type CAN_ID_32.	
Template Description		
M2 Parameter	SystemTemplate::TransportProtocols::CanTpConnection.flowControlPdu	
Mapping Rule	Mapping Type	
Create container if the CanTpConnection contains a reference to a FlowControl NPdu that is received by the regarded ECU.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00271]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpRxFcNPdu	
BSW Parameter	BSW Type	
CanTpRxFcNPduld	EcucIntegerParamDef	
BSW Description	N-PDU identifier attached to the FC N-PDU of this TxNSdu identified by CanTpTxNSduld.	
Each TxNsdu identifier is linked to one Rx FC N-PDU identifier only. However, in the case of extended addressing format, the same FC N-PDU identifier can be used for several N-SDU identifiers. The distinction is made by means of the N_TA value (first data byte of FC frames).		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00273]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpRxFcNPdu	
BSW Parameter	BSW Type	
CanTpRxFcNPduRef	EcucReferenceDef	
BSW Description	Reference to a Pdu in the COM-Stack.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00272]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter	BSW Type	
CanTpTc	EcucBooleanParamDef	
BSW Description	Switch for enabling Transmit Cancellation.	
Template Description	With this switch Tx Cancellation can be turned on or off. Please note that the Rx Cancellation is always enabled.	
M2 Parameter	SystemTemplate::TransportProtocols::CanTpConnection.cancellation	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00282]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter	BSW Type	
CanTpTxAddressingFormat	EcucEnumerationParamDef	
BSW Description	Declares which communication addressing format is supported for this TxNSdu. Definition of Enumeration values: CanTpStandard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.	
Template Description	Declares which communication addressing mode is supported.	
M2 Parameter	SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00262]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter	BSW Type	
CanTpTxNPdu	EcucParamConfContainerDef	
BSW Description	Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description		
M2 Parameter	SystemTemplate::TransportProtocols::CanTpConnection.dataPdu	
Mapping Rule	Mapping Type	
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00274]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNPdu	
BSW Parameter	BSW Type	
CanTpTxNPduConfirmationPduld	EcucIntegerParamDef	
BSW Description	Handle Id to be used by the CanIf to confirm the transmission of the CanTpTxNPdu to the CanIf module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00286]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNPdu	
BSW Parameter	BSW Type	
CanTpTxNPduRef	EcucReferenceDef	
BSW Description	Reference to a Pdu in the COM-Stack.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTp_00275]	

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	

BSW Parameter		BSW Type
CanTpTxNSduId		EcucIntegerParamDef
BSW Description		
Unique identifier to a structure that contains all useful information to process the transmission of a TxNsdu.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00268]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpTxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTp_00261]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpTxPaddingActivation		EcucEnumerationParamDef
BSW Description		
Defines if the transmit frame use padding or not. This parameter is restricted to 8 byte N-PDUs.		
Definition of Enumeration values:		
CanTpOn The transmit N-PDU uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes in case of CAN 2.0)		
CanTpOff The transmit N-PDU does not use padding for SF, CF and the last CF. (N-PDU length is dynamic - any valid DLC value).		
Note: The mandatory mapping to the next higher valid DLC value for N-PDUs with a length > 8 bytes is not affected by this parameter.		
Template Description		

This specifies wheter or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.	
true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes)	
false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.paddingActivation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00269]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	BSW Type
CanTpTxTaType	EcucEnumerationParamDef
BSW Description	
Declares the communication type of this TxNsdu.	
Enumeration values: CanTpPhysical. Used for 1:1 communication. CanTpFunctional. Used for 1:n communication.	
Template Description	
Network Target Address type.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.taType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00270]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig
BSW Parameter	BSW Type
CanTpMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Allow to configure the time for the MainFunction (as float in seconds). The CanTpMainFunctionPeriod should be assigned a value which is optimal regarding all of the timers configured for CanTp in TX and RX data transfer i.e. the differences from the configured timing should be as small as possible. Please note: This period shall be the same as call cycle time of the periodic task were CanTp Main function is called.	
Template Description	
The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpEcu.cycleTimeMainFunction	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00240]

BSW Module		BSW Context	
CanTp		CanTp/CanTpConfig	
BSW Parameter		BSW Type	
CanTpMaxChannelCnt		EcucIntegerParamDef	
BSW Description			
Maximum number of channels. This parameter is needed only in case of post-build loadable implementation using static memory allocation.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00304]	

BSW Module		BSW Context	
CanTp		CanTp	
BSW Parameter		BSW Type	
CanTpGeneral		EcucParamConfContainerDef	
BSW Description			
This container contains the general configuration parameters of the CanTp module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00278]	

BSW Module		BSW Context	
CanTp		CanTp/CanTpGeneral	
BSW Parameter		BSW Type	
CanTpChangeParameterApi		EcucBooleanParamDef	
BSW Description			
This parameter, if set to true, enables the CanTp_ChangeParameterRequest Api for this Module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTp_00299]	

BSW Module		BSW Context	
CanTp		CanTp/CanTpGeneral	
BSW Parameter		BSW Type	
CanTpDevErrorDetect		EcucBooleanParamDef	

BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00239]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpDynIdSupport	EcucBooleanParamDef
BSW Description	
Enable support for dynamic ID handling via N-PDU MetaData.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00302]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpFlexibleDataRateSupport	EcucBooleanParamDef
BSW Description	
Enable support for CAN FD frames.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00305]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpGenericConnectionSupport	EcucBooleanParamDef
BSW Description	
Enable support for the handling of generic connections using N-SDUs with MetaData. Requires CanTpDynIdSupport.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00303]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpPaddingByte	EcucIntegerParamDef
BSW Description	
Used for the initialization of unused bytes with a certain value	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00298]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpReadParameterApi	EcucBooleanParamDef
BSW Description	
This parameter, if set to true, enables the CanTp_ReadParameterApi for this module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00300]

BSW Module	BSW Context
CanTp	CanTp/CanTpGeneral
BSW Parameter	BSW Type
CanTpVersionInfoApi	EcucBooleanParamDef
BSW Description	
The function CanTp_GetVersionInfo is configurable (On/Off) by this configuration parameter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00283]

C.2.6 CanSm Mapping

BSW Module	BSW Context	
CanSM	CanSM	
BSW Parameter	BSW Type	
CanSMConfiguration	EcucParamConfContainerDef	
BSW Description		
This container contains the global parameters of the CanSM and sub containers, which are for the CAN network specific configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00123]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration	
BSW Parameter	BSW Type	
CanSMManagerNetwork	EcucParamConfContainerDef	
BSW Description		
This container contains the CAN network specific parameters of each CAN network		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00126]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMBorCounterL1ToL2	EcucIntegerParamDef	
BSW Description		
This threshold defines the count of bus-offs until the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).		
Template Description		
This threshold defines the count of bus-offs until the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanClusterBusOffRecovery.borCounterL1ToL2		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanSM_00131]

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMBorTimeL1	EcucFloatParamDef	
BSW Description		
This time parameter defines in seconds the duration of the bus-off recovery time in level 1 (short recovery time).		
Template Description		
This attribute defines the duration of the bus-off recovery time in level 1 (short recovery time) in seconds.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanClusterBusOffRecovery.borTimeL1		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00128]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMBorTimeL2	EcucFloatParamDef	
BSW Description		
This time parameter defines in seconds the duration of the bus-off recovery time in level 2 (long recovery time).		
Template Description		
This attribute defines the duration of the bus-off recovery time in level 2 (long recovery time) in seconds.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanClusterBusOffRecovery.borTimeL2		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00129]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMBorTimeTxEnsured	EcucFloatParamDef	
BSW Description		
This parameter defines in seconds the duration of the bus-off event check. This check assesses, if the recovery has been successful after the recovery reenables the transmit path. If a new bus-off occurs during this time period, the CanSM assesses this bus-off as sequential bus-off without successful recovery. Because a bus-off only can be detected, when PDUs are transmitted, the time has to be great enough to ensure that PDUs are transmitted again (e. g. time period of the fastest cyclic transmitted PDU of the COM module / ComTxModeTimePeriodFactor).		
Template Description		
This attribute defines the duration of the bus-off event check in seconds.		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanClusterBusOffRecovery.borTimeTxEnsured	
Mapping Rule	Mapping Type
If borTimeTxEnsured is defined set this parameter to true otherwise to false.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanSM_00130]

BSW Module	BSW Context
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork
BSW Parameter	BSW Type
CanSMBorTxConfirmationPolling	EcucBooleanParamDef
BSW Description	
This parameter shall configure, if the CanSM polls the CanIf_GetTxConfirmationState API to decide the bus-off state to be recovered instead of using the CanSMBorTimeTxEnsured parameter for this decision.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanSM_00339]

BSW Module	BSW Context
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork
BSW Parameter	BSW Type
CanSMComMNetworkHandleRef	EcucReferenceDef
BSW Description	
Unique handle to identify one certain CAN network. Reference to one of the network handles configured for the ComM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanSM_00161]

BSW Module	BSW Context
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork
BSW Parameter	BSW Type
CanSMController	EcucParamConfContainerDef
BSW Description	
This container contains the controller IDs assigned to a CAN network.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00338]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork/CanSMController	
BSW Parameter		BSW Type
CanSMControllerId		EcucReferenceDef
BSW Description		
Unique handle to identify one certain CAN controller. Reference to one of the CAN controllers managed by the CanIf module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00141]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter		BSW Type
CanSMDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00127]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork/CanSMDemEventParameterRefs	
BSW Parameter		BSW Type
CANSM_E_BUS_OFF		EcucReferenceDef
BSW Description		
Reference to configured DEM event to report bus off errors for this CAN network.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00070]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork/CanSMDemEventParameterRefs	
BSW Parameter	BSW Type	
CANSM_E_MODE_REQUEST_TIMEOUT	EcucReferenceDef	
BSW Description		
Reference to configured DEM event to report bus off errors for this CAN network.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00352]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMEnableBusOffDelay	EcucBooleanParamDef	
BSW Description		
This parameter defines if the <User_GetBusOffDelay> shall be called for this network.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanSM_00346]	

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter	BSW Type	
CanSMTransceiverId	EcucReferenceDef	
BSW Description		
ID of the CAN transceiver assigned to the configured network handle. Reference to one of the transceivers managed by the CanIf module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_CanSM_00137]
-------	--------------------

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration	
BSW Parameter	BSW Type	
CanSMModeRequestRepetitionMax	EcucIntegerParamDef	
BSW Description		
Specifies the maximal amount of mode request repetitions without a respective mode indication from the CanIf module until the CanSM module reports a Development Error to the Det and tries to go back to no communication.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanSM_00335]

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration	
BSW Parameter	BSW Type	
CanSMModeRequestRepetitionTime	EcucFloatParamDef	
BSW Description		
Specifies in which time duration the CanSM module shall repeat mode change requests by using the API of the CanIf module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanSM_00336]

BSW Module	BSW Context	
CanSM	CanSM	
BSW Parameter	BSW Type	
CanSMGeneral	EcucParamConfContainerDef	
BSW Description		
Container for general pre-compile parameters of the CanSM module		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanSM_00314]

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00133]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMGetBusOffDelayFunction		EcucFunctionNameDef	
BSW Description			
This parameter configures the name of the <User_GetBusOffDelay> callout function, which is used by CanSM to acquire an additional L1/L2 delay time. This function is only called for channels where CanSMEnableBusOffDelay is enabled.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00347]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMGetBusOffDelayHeader		EcucStringParamDef	
BSW Description			
This parameter configures the header file containing the prototype of the <User_GetBusOffDelay> callout function.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00348]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMIcomSupport		EcucBooleanParamDef	
BSW Description			
Selects support of Pretended Network features in CanSM. True: Enabled False: Disabled			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00345]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMMainFunctionTimePeriod		EcucFloatParamDef	
BSW Description			
This parameter defines the cycle time of the function CanSM_MainFunction in seconds			
Template Description			
This attribute defines the cycle time of the function CanSM_MainFunction in seconds.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanClusterBusOffRecovery.mainFunctionPe- riod			
Mapping Rule		Mapping Type	
The value that is defined in the System Extract defines the upperbound of the cycle time. The integrator may choose a smaller value.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00312]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMPncSupport		EcucBooleanParamDef	
BSW Description			
Enables or disables support of partial networking. False: Partial Networking is disabled True: Partial Networking is enabled			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00344]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMSetBaudrateApi		EcucBooleanParamDef	
BSW Description			
The support of the Can_SetBaudrate API is optional. If this parameter is set to true the Can_SetBaudrate API shall be supported. Otherwise the API is not supported.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00343]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMTxOfflineActiveSupport		EcucBooleanParamDef	
BSW Description			
Determines whether the ECU passive feature is supported by CanSM. True: Enabled False: Disabled			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00349]	

BSW Module		BSW Context	
CanSM		CanSM/CanSMGeneral	
BSW Parameter		BSW Type	
CanSMVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Activate/Deactivate the version information API (CanSM_GetVersionInfo). true: version information API activated false: version information API deactivated			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanSM_00311]	

C.3 J1939

C.3.1 J1939Tp Mapping

BSW Module	BSW Context	
J1939Tp	J1939Tp	
BSW Parameter	BSW Type	
J1939TpConfiguration	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters and sub containers of the J1939Tp module that define the communication paths.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00052]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration	
BSW Parameter	BSW Type	
J1939TpRxChannel	EcucParamConfContainerDef	
BSW Description		
This container describes a reception channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.		
Template Description		
A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection		
Mapping Rule		Mapping Type
Create container for each existing J1939TpConnection that is used to transmit a NSdu.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00053]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxCancellationSupport	EcucBooleanParamDef	
BSW Description		
Enable receive cancellation using the API J1939Tp_CancelReceive() for this channel.		
Template Description		
Enable support for Tx/Rx cancellation.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation		

Mapping Rule		Mapping Type
Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00186]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxCmNPdu	EcucParamConfContainerDef	
BSW Description		
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection. This N-PDU consumes a meta data item of type CAN_ID_32.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu		
Mapping Rule	Mapping Type	
Information can be derived from a received directINPdu that is referenced by the J1939TpConnection.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00128]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxCmNPdu	
BSW Parameter	BSW Type	
J1939TpRxCmNPduld	EcucIntegerParamDef	
BSW Description		
The N-PDU identifier used for communication with CanIf.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00129]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxCmNPdu	
BSW Parameter	BSW Type	
J1939TpRxCmNPduRef	EcucReferenceDef	
BSW Description		
Reference to the Pdu object representing the N-PDU.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00158]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpRxDa		EcucIntegerParamDef
BSW Description		
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00178]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpRxDtNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU. This N-PDU consumes a meta data item of type CAN_ID_32.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dataPdu		
Mapping Rule		Mapping Type
Information can be derived from a received NPdu that is referenced by the J1939 TpConnection.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00117]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxDtNPdu	
BSW Parameter		BSW Type
J1939TpRxDtNPdul		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for communication with Canlf.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00133]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxDtNPdu	
BSW Parameter	BSW Type	
J1939TpRxDtNPduRef	EcucReferenceDef	
BSW Description		
Reference to the Pdu object representing the N-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00134]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxDynamicBlockCalculation	EcucBooleanParamDef	
BSW Description		
Enable dynamic calculation of "number of packets that can be sent" value in TP.CM_CTS, based on the size of buffers in upper layers reported via StartOfReception and PduR_J1939TpCopyRxData.		
Template Description		
Enable support for dynamic block size calculation.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dynamicBs		
Mapping Rule		Mapping Type
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an EC U shall have the same value.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00187]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxDynamicBufferRatio	EcucIntegerParamDef	
BSW Description		
Percentage of available buffer that shall be used for retry. This parameter is only applicable when "J1939TpRxRetrySupport" and "J1939TpRxDynamicBlockCalculation" are enabled.		
Template Description		
Defines usage of available data for dynamic block size calculation when protocol retry is enabled. This attribute describes in percent of available buffer that shall be used for retry.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.bufferRatio		

Mapping Rule		Mapping Type
Please note that in the System Template this attribute is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00188]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxPacketsPerBlock	EcucIntegerParamDef	
BSW Description		
Number of TP.DT frames the receiving J1939Tp module allows the sender to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_CTS frame, and is thus only relevant for reception of messages via CMDT. When J1939TpRxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.		
Template Description		
Set maximum block size (number of packets in TP.CM_CTS).		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.maxBs		
Mapping Rule		Mapping Type
Please note that in the System Template the maximum block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00189]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter	BSW Type	
J1939TpRxPg	EcucParamConfContainerDef	
BSW Description		
Parameter group received by the J1939 transport layer.		
Template Description		
A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg		
Mapping Rule		Mapping Type
Create container for each Rx J1939TpPg that is available in the Ecu Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00050]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter	BSW Type	
J1939TpRxDirectNPdu	EcucParamConfContainerDef	
BSW Description		

This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes. This N-PDU consumes a meta data item of type CAN_ID_32.	
Please note: This sub container is only necessary when J1939TpRxPgDynLength is TRUE.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpPg.directPdu	
Mapping Rule	Mapping Type
Information can be derived from a received directNPdu that is referenced by the J1939TpPg.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00130]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxDirectNPdu
BSW Parameter	BSW Type
J1939TpRxDirectNPdulid	EcucIntegerParamDef
BSW Description	
The N-PDU identifier used for communication with CanIf.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00131]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxDirectNPdu
BSW Parameter	BSW Type
J1939TpRxDirectNPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the N-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00132]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg
BSW Parameter	BSW Type
J1939TpRxNSdu	EcucParamConfContainerDef

BSW Description	
This container describes the parameters that are relevant for the reception of a specific N-SDU. This N-SDU produces meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpPg.sdu	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00063]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxNSdu
BSW Parameter	BSW Type
J1939TpRxNSdulid	EcucIntegerParamDef
BSW Description	
This is a unique identifier for a received N-SDU. This Id is used in the CancelReceive and ChangeParameter API call.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00184]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxNSdu
BSW Parameter	BSW Type
J1939TpRxNSduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the N-SDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00069]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg
BSW Parameter	BSW Type
J1939TpRxPgDynLength	EcucBooleanParamDef

BSW Description	
This flag is set to TRUE when the N-SDU refers to a PGN with variable length.	
Please note: When this attribute is TRUE, the sub container J1939TpRxDirectNPdu is required.	
Template Description	
The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SystemSignal.dynamicLength	
Mapping Rule	Mapping Type
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00066]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter		BSW Type
J1939TpRxPgPGN		EcucIntegerParamDef
BSW Description		
PGN of the referenced N-SDUs.		
Template Description		
Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg.pgn		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00065]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpRxProtocolType		EcucEnumerationParamDef
BSW Description		
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.		
Template Description		
BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.broadcast		
Mapping Rule		Mapping Type
If the broadcast attribute is set to true than set this parameter to J1939TP_PRO TOCOL_BAM		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00029]

BSW Module	BSW Context
-------------------	--------------------

J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpRxRetrySupport	EcucBooleanParamDef	
BSW Description		
Enable support for triggering repetition of failed transmission using TPCM_CTS with a packet number that has already been sent. Retransmission is triggered when a sequence number is missing or a timeout occurs during reception.		
Template Description		
Enable support for protocol retry.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.retry		
Mapping Rule		Mapping Type
Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00185]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpRxSa	EcucIntegerParamDef	
BSW Description		
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with MetaData containing the SA.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Tp_00179]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		BSW Type
J1939TpTxFcNPdu	EcucParamConfContainerDef	
BSW Description		
This N-PDU represents the TPCM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TPCM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. This N-PDU produces a meta data item of type CAN_ID_32.		
Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu		
Mapping Rule		Mapping Type
Information can be derived from a received FlowControlNPdu that is referenced by the J1939TpConnection.		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00135]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpTxFcNPdu	
BSW Parameter		BSW Type
J1939TpTxFcNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU.		
Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpTxCmNPduRef of the corresponding J1939TpTxChannel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00136]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpTxFcNPdu	
BSW Parameter		BSW Type
J1939TpTxFcNPduTxConfId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for Tx confirmation from CanIf.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00168]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration	
BSW Parameter		BSW Type
J1939TpTxChannel		EcucParamConfContainerDef
BSW Description		
This container describes a transmission channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.		
Template Description		
A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.		
M2 Parameter		

SystemTemplate::TransportProtocols::J1939TpConnection	
Mapping Rule	Mapping Type
Create container for each existing J1939TpConnection that is used to transmit a NSdu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00059]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpRxFcNPdu	EcucParamConfContainerDef
BSW Description	
This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. This N-PDU consumes a meta data item of type CAN_ID_32.	
Please note: This sub container is only required when J1939TpTxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu	
Mapping Rule	Mapping Type
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939TpConnection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00144]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpRxFcNPdu
BSW Parameter	BSW Type
J1939TpRxFcNPdul	EcucIntegerParamDef
BSW Description	
The N-PDU identifier used for communication with CanIf.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00145]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpRxFcNPdu
BSW Parameter	BSW Type
J1939TpRxFcNPduRef	EcucReferenceDef
BSW Description	

Reference to the Pdu object representing the N-PDU.	
Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpRxCmNPduRef of the corresponding J1939TpRxChannel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00146]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpTxCancellationSupport	EcucBooleanParamDef
BSW Description	
Enable transmit cancellation using the API J1939Tp_CancelTransmit() for this channel.	
Template Description	
Enable support for Tx/Rx cancellation.	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation	
Mapping Rule	Mapping Type
Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00192]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpTxCmNPdu	EcucParamConfContainerDef
BSW Description	
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu	
Mapping Rule	Mapping Type
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939TpConnection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00138]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxCmNPdu

BSW Parameter		BSW Type	
J1939TpTxCmNPduRef		EcucReferenceDef	
BSW Description			
Reference to the Pdu object representing the N-PDU.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00139]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxCmNPdu	
BSW Parameter		BSW Type	
J1939TpTxCmNPduTxConfId		EcucIntegerParamDef	
BSW Description			
The N-PDU identifier used for Tx confirmation from Canlf.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00170]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		BSW Type	
J1939TpTxDa		EcucIntegerParamDef	
BSW Description			
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00180]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		BSW Type	
J1939TpTxDtNPdu		EcucParamConfContainerDef	
BSW Description			

This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.dataPdu	
Mapping Rule	Mapping Type
Information can be derived from a transmitted NPdu that is referenced by the J1939TpConnection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00142]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxDtNPdu
BSW Parameter	BSW Type
J1939TpTxDtNPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the N-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00143]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxDtNPdu
BSW Parameter	BSW Type
J1939TpTxDtNPduTxConfId	EcucIntegerParamDef
BSW Description	
The N-PDU identifier used for Tx confirmation from CanIf.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00171]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpTxDynamicBlockCalculation	EcucBooleanParamDef
BSW Description	
Enable dynamic calculation of "maximum number of packets that can be sent" value in TP.CM_RTS, based on the available amount of data in upper layers reported via PduR_J1939TpCopyTxData.	

Template Description	
Enable support for dynamic block size calculation.	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.dynamicBs	
Mapping Rule	Mapping Type
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an EC U shall have the same value.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00191]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpTxMaxPacketsPerBlock	EcucIntegerParamDef
BSW Description	
Maximum number of TP.DT frames the transmitting J1939Tp module is ready to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_RTS frame, and is thus only relevant for transmission of messages via CMDT. When J1939TpTxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.	
Template Description	
Set maximum for expected block size (maximum number of packets in TP.CM_RTS).	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.maxExpBs	
Mapping Rule	Mapping Type
Please note that in the System Template the maximum for expected block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00190]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	BSW Type
J1939TpTxPg	EcucParamConfContainerDef
BSW Description	
Parameter group transmitted by the J1939 transport layer.	
Template Description	
A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpPg	
Mapping Rule	Mapping Type
Create container for each Tx J1939TpPg that is available in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00070]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	BSW Type

J1939TpTxDirectNPdu	EcucParamConfContainerDef
BSW Description	
This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less than 8 bytes. This N-PDU produces a meta data item of type CAN_ID_32.	
Please note: This sub container is only necessary when J1939TpTxPgDynLength is TRUE.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpPg.directPdu	
Mapping Rule	Mapping Type
Information can be derived from a transmitted directNPdu that is referenced by the J1939TpPg.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00140]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxDirectNPdu
BSW Parameter	BSW Type
J1939TpTxDirectNPduRef	EcucReferenceDef
BSW Description	
Reference to the Pdu object representing the N-PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00141]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxDirectNPdu
BSW Parameter	BSW Type
J1939TpTxDirectNPduTxConfId	EcucIntegerParamDef
BSW Description	
The N-PDU identifier used for Tx confirmation from CanIf.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00169]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg

BSW Parameter		BSW Type	
J1939TpTxNSdu		EcucParamConfContainerDef	
BSW Description			
This container describes the parameters that are relevant for the transmission of a specific N-SDU. This N-SDU consumes meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8.			
Template Description			
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpPg.sdu			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00147]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxNSdu	
BSW Parameter		BSW Type	
J1939TpTxNSduId		EcucIntegerParamDef	
BSW Description			
The N-SDU identifier used for communication with PduR.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00149]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxNSdu	
BSW Parameter		BSW Type	
J1939TpTxNSduRef		EcucReferenceDef	
BSW Description			
Reference to the Pdu object representing the N-SDU.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00151]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg	
BSW Parameter		BSW Type	

J1939TpTxPgDynLength	EcucBooleanParamDef
BSW Description	
This flag is set to TRUE when the N-SDU refers to a PGN with variable length.	
Please note: When this attribute is TRUE, the sub container J1939TpTxDirectNPdu is required.	
Template Description	
The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::SystemSignal.dynamicLength	
Mapping Rule	Mapping Type
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Tp_00148]

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg	
BSW Parameter	BSW Type	
J1939TpTxPgPGN	EcucIntegerParamDef	
BSW Description		
PGN of the referenced N-SDUs.		
Template Description		
Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg.pgn		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00150]	

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter	BSW Type	
J1939TpTxProtocolType	EcucEnumerationParamDef	
BSW Description		
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.		
Template Description		
BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.		
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.broadcast		
Mapping Rule	Mapping Type	
If the broadcast attribute is set to true than set this parameter to J1939TP_PRO TOCOL_BAM	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Tp_00137]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		BSW Type	
J1939TpTxRetrySupport		EcucBooleanParamDef	
BSW Description			
Enable support for repetition of failed transmission using TP.CM_CTS with a packet number that has already been sent. Retransmission is handled via the retry feature of PduR_J1939TpCopyTxData.			
Template Description			
Enable support for protocol retry.			
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpConnection.retry			
Mapping Rule			Mapping Type
Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00193]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		BSW Type	
J1939TpTxSa		EcucIntegerParamDef	
BSW Description			
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with MetaData containing the SA.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00181]	

BSW Module		BSW Context	
J1939Tp		J1939Tp	
BSW Parameter		BSW Type	
J1939TpGeneral		EcucParamConfContainerDef	
BSW Description			
This container describes the general configuration parameters of the J1939Tp module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00033]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpGeneral	

BSW Parameter		BSW Type	
J1939TpCancellationSupport		EcucBooleanParamDef	
BSW Description			
Enable transmit and receive cancellation. The APIs J1939Tp_CancelTransmit() and J1939Tp_CancelReceive() will only be available when this parameter is enabled.			
Template Description			
Enable support for Tx/Rx cancellation.			
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation			
Mapping Rule		Mapping Type	
Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00174]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpGeneral	
BSW Parameter		BSW Type	
J1939TpDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00042]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpGeneral	
BSW Parameter		BSW Type	
J1939TpMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
Allow to configure the time for the MainFunction (in seconds). Please note: This configuration value shall be equal to the value in the ScheduleManager module.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00044]	

BSW Module		BSW Context	
J1939Tp		J1939Tp/J1939TpGeneral	
BSW Parameter		BSW Type	
J1939TpVersionInfoApi		EcucBooleanParamDef	
BSW Description			
The function J1939Tp_GetVersionInfo is configurable (On/Off) by this configuration parameter.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Tp_00051]	

C.3.2 J1939Nm Mapping

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet	
BSW Parameter		BSW Type	
J1939NmChannel		EcucParamConfContainerDef	
BSW Description			
Physical CAN channel handled by J1939Nm.			
Template Description			
J1939 specific NmCluster attributes			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NmCluster			
Mapping Rule			Mapping Type
Create Container for each existing J1939NmCluster.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00005]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmChannel	
BSW Parameter		BSW Type	
J1939NmChannelUsesAddressArbitration		EcucBooleanParamDef	
BSW Description			
Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes.			
True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not react to other address claims.			
Template Description			
Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes.			
True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not care for contending address claims.			

M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.usesAddressArbitration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00035]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel
BSW Parameter	BSW Type
J1939NmRxPdu	EcucParamConfContainerDef
BSW Description	
Contains the configuration of the PDU used to receive the AddressClaimed PG. This PDU consumes a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.rxNmPdu	
Mapping Rule	Mapping Type
Shall be derived from the NmPdu that is referenced by the NmNode.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00010]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel
BSW Parameter	BSW Type
J1939NmTxPdu	EcucParamConfContainerDef
BSW Description	
Contains the configuration of the PDU used to transmit the AddressClaimed PG. This PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.txNmPdu	
Mapping Rule	Mapping Type
Shall be derived from the NmPdu that is referenced by the NmNode.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00009]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	BSW Type
J1939NmExternalNodeNameArbitraryAddressCapable	EcucBooleanParamDef
BSW Description	
Arbitrary Address Capable field of the NAME of this external node.	
Template Description	
Arbitrary Address Capable field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.arbitraryAddressCapable	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00041]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type
J1939NmExternalNodeNameECUInstance		EcucIntegerParamDef
BSW Description		
ECU Instance field of the NAME of this external node.		
Template Description		
ECU Instance field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.ecuInstance		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00042]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type
J1939NmExternalNodeNameFunction		EcucIntegerParamDef
BSW Description		
Function field of the NAME of this external node.		
Template Description		
Function field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.function		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00043]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type
J1939NmExternalNodeNameFunctionInstance		EcucIntegerParamDef
BSW Description		
Function Instance field of the NAME of this external node.		
Template Description		
Function Instance field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.functionInstance		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00044]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type	
J1939NmExternalNodeNameIdentityNumber		EcucIntegerParamDef	
BSW Description			
Identity Number field of the NAME of this external node.			
Template Description			
Identity Number field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.identityNumber			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00045]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type	
J1939NmExternalNodeNameIndustryGroup		EcucIntegerParamDef	
BSW Description			
Industry Group field of the NAME of this external node.			
Template Description			
Industry Group field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.industryGroup			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00046]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type	
J1939NmExternalNodeNameManufacturerCode		EcucIntegerParamDef	
BSW Description			
Manufacturer Code field of the NAME of this external node.			
Template Description			
Manufacturer Code field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.manufacturerCode			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00047]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		BSW Type	
J1939NmExternalNodeNameVehicleSystem		EcucIntegerParamDef	
BSW Description			

Vehicle System field of the NAME of this external node.	
Template Description	
Vehicle System field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.vehicleSystem	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00048]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	BSW Type
J1939NmExternalNodeNameVehicleSystemInstance	EcucIntegerParamDef
BSW Description	
Vehicle System Instance field of the NAME of this external node.	
Template Description	
Vehicle System Instance field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.vehicleSystemInstance	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00050]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	BSW Type
J1939NmExternalNodePreferredAddress	EcucIntegerParamDef
BSW Description	
Source address of this external node.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmNodeId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00049]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	BSW Type
J1939NmNodeChannelRef	EcucReferenceDef
BSW Description	
Reference to the channels this node has access to.	
Template Description	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmNode	

Mapping Rule		Mapping Type
This reference shall be derived from NmClusters that aggregate the nmNode in the Ecu Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00029]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type
J1939NmNodeNameArbitraryAddressCapable		EcucBooleanParamDef
BSW Description		
Arbitrary Address Capable field of the NAME of this node.		
Template Description		
Arbitrary Address Capable field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.arbitraryAddressCapable		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00018]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type
J1939NmNodeNameECUInstance		EcucIntegerParamDef
BSW Description		
ECU Instance field of the NAME of this node.		
Template Description		
ECU Instance field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.ecuInstance		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00024]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type
J1939NmNodeNameFunction		EcucIntegerParamDef
BSW Description		
Function field of the NAME of this node.		
Template Description		
Function field of the NAME of this node.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NodeName.function		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00022]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type	
J1939NmNodeNameFunctionInstance		EcucIntegerParamDef	
BSW Description			
Function Instance field of the NAME of this node.			
Template Description			
Function Instance field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.functionInstance			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00023]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type	
J1939NmNodeNameIdentityNumber		EcucIntegerParamDef	
BSW Description			
Identity Number field of the NAME of this node.			
Template Description			
Identity Number field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.identityNumber			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00026]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type	
J1939NmNodeNameIndustryGroup		EcucIntegerParamDef	
BSW Description			
Industry Group field of the NAME of this node.			
Template Description			
Industry Group field of the NAME of this node.			
M2 Parameter			
SystemTemplate::NetworkManagement::J1939NodeName.industryGroup			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_J1939Nm_00019]	

BSW Module		BSW Context	
J1939Nm		J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		BSW Type	
J1939NmNodeNameManufacturerCode		EcucIntegerParamDef	
BSW Description			

Manufacturer Code field of the NAME of this node.	
Template Description	
Manufacturer Code field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.manufacturerCode	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00025]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	BSW Type
J1939NmNodeNameVehicleSystem	EcucIntegerParamDef
BSW Description	
Vehicle System field of the NAME of this node.	
Template Description	
Vehicle System field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.vehicleSystem	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00021]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	BSW Type
J1939NmNodeNameVehicleSystemInstance	EcucIntegerParamDef
BSW Description	
Vehicle System Instance field of the NAME of this node.	
Template Description	
Vehicle System Instance field of the NAME of this node.	
M2 Parameter	
SystemTemplate::NetworkManagement::J1939NodeName.vehicleSystemInstance	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00020]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	BSW Type
J1939NmNodePreferredAddress	EcucIntegerParamDef
BSW Description	
Source address of this node used for address claiming.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmNodeId	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00016]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet	
BSW Parameter		BSW Type
J1939NmSharedAddressSpace		EcucParamConfContainerDef
BSW Description		
Set of J1939NmChannels that share a common address space. Address claims will be routed between these channels.		
Template Description		
This meta-class represents the ability to identify several J1939Clusters that share a common address space for the routing of messages		
M2 Parameter		
SystemTemplate::J1939SharedAddressCluster		
Mapping Rule		Mapping Type
Container shall be created for each existing J1939SharedAddressCluster		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00037]	

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmSharedAddressSpace	
BSW Parameter		BSW Type
J1939NmSharedChannelRef		EcucReferenceDef
BSW Description		
Reference to a channel that belongs to the shared address space.		
Template Description		
This identifies the J1939Clusters that share a common address space		
M2 Parameter		
SystemTemplate::J1939SharedAddressCluster.participatingJ1939Cluster		
Mapping Rule		Mapping Type
Reference shall be created for each J1939 cluster that is referenced by J1939 SharedAddressCluster in the role participating1939Cluster.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Nm_00038]	

C.3.3 J1939Dcm Mapping

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode	
BSW Parameter		BSW Type
J1939DcmDiagnosticMessageSupport		EcucParamConfContainerDef
BSW Description		
Contains parameters to configure the diagnostic message support		
Template Description		
Represents the IPdus handled by J1939Dcm.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::J1939DcmIPdu		

Mapping Rule		Mapping Type
The container shall be created for every J1939DcmIPdu that is transmitted oder received by the regarded Ecu.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Dcm_00014]	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport	
BSW Parameter		BSW Type
J1939DcmDmxSupport		EcucEnumerationParamDef
BSW Description		
This parameter is used to identify the actual DMx message.		
Template Description		
This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::J1939DcmIPdu.diagnosticMessageType		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Dcm_00042]	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport	
BSW Parameter		BSW Type
J1939DcmRxPdu		EcucParamConfContainerDef
BSW Description		
Contains parameters to configure the J1939DcmRxPdu.		
This PDU consumes meta data items of type CAN_ID_32 for PDUs received from CanIf, and of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8 for PDUs received from J1939Tp.		
Template Description		
Represents the IPdus handled by J1939Dcm.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::J1939DcmIPdu		
Mapping Rule		Mapping Type
The direction of the J1939DcmIPdu shall be derived from the PduTriggering and the references to IPduPorts.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_J1939Dcm_00046]	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport	
BSW Parameter		BSW Type
J1939DcmTxPdu		EcucParamConfContainerDef
BSW Description		

Contains parameters to configure the J1939DcmTxPdu.	
This PDU produces meta data items of type CAN_ID_32 for PDUs transmitted via CanIf, and of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8 for PDUs transmitted via J1939Tp.	
Template Description	
Represents the IPdus handled by J1939Dcm.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::J1939DcmIPdu	
Mapping Rule	Mapping Type
The direction of the J1939DcmIPdu shall be derived from the PduTriggering and the references to IPduPorts.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Dcm_00045]

C.3.4 J1939Rm Mapping

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet	
BSW Parameter		BSW Type
J1939RmChannel		EcucParamConfContainerDef
BSW Description		
Contains the parameters for a CAN channel supported by the J1939 Request Manager.		
Template Description		
J1939 specific NmCluster attributes		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NmCluster		
Mapping Rule		Mapping Type
Container shall be created for each J1939NmCluster that is available in the Ecu Extract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Rm_00009]

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmChannel	
BSW Parameter		BSW Type
J1939RmRqst2RxPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration of the I-PDU used to receive the Request2 PG. This PDU consumes a meta data item of type CAN_ID_32.		
Template Description		
Enables support for the Request2 PGN (RQST2).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.request2Support		
Mapping Rule		Mapping Type
Create container if J1939Cluster.request2Support is set to true.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Rm_00075]

BSW Module	BSW Context
-------------------	--------------------

J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmChannel	
BSW Parameter		BSW Type
J1939RmRqst2TxPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration of the I-PDU used to transmit the Request2 PG. This PDU produces a meta data item of type CAN_ID_32.		
Template Description		
Enables support for the Request2 PGN (RQST2).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.request2Support		
Mapping Rule		Mapping Type
Create container if J1939Cluster.request2Support is set to true.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Rm_00076]

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet	
BSW Parameter		BSW Type
J1939RmNode		EcucParamConfContainerDef
BSW Description		
Contains the parameters for the support of a logical J1939 node (identified by an ECU address).		
Template Description		
J1939 specific NM Node attributes.		
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NmNode		
Mapping Rule		Mapping Type
J1939RmNode shall be derived from existing J1939NmNodes that are available in the ExuExtract. Please note that J1939NmNodes that have the same short Name and nmNodeId that are located on different NmClusters shall be combined to one J1939RmNode.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Rm_00049]

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode	
BSW Parameter		BSW Type
J1939RmNodeChannelRef		EcucReferenceDef
BSW Description		
Reference to the channels this node has access to.		
Template Description		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNode		
Mapping Rule		Mapping Type
This reference shall be derived from NmClusters that aggregate the nmNode in the Ecu Extract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_J1939Rm_00052]

BSW Module	BSW Context
-------------------	--------------------

J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode
BSW Parameter	
J1939RmUser	EcucChoiceContainerDef
BSW Description	
Contains the configuration of a module that uses the request and acknowledgement interfaces of J1939Rm.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
+ J1939NM user exists always, UserPGN has 0x0ee00 as solitary value + J1939DCM user exists if transmitted J1939DcmIPdus exist which are requestable + COM user exists if transmitted ISignallPdus exist which are requestable + CDD user exists if transmitted UserDefinedPdus or UserDefinedIPdus exist which are requestable + RTE users cannot be derived	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00010]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmCdd User
BSW Parameter	
J1939RmUserRequestPGN	EcucIntegerParamDef
BSW Description	
PGN supported to be requested from this module. The PGNs supported by different modules should usually be disjunctive.	
Template Description	
<p>Pdu: Collection of all Pdus that can be routed through a bus interface.</p> <p>CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message.</p> <p>CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network.</p> <p>J1939TpPg.requestable: Parameter Group can be triggered by the J1939 request message.</p> <p>J1939TpPg.pgn: Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu, SystemTemplate::Fibex::Fibex4 Can::CanCommunication::CanFrameTriggering.j1939requestable, SystemTemplate::Fibex::Fibex4 Can::CanCommunication::CanFrameTriggering.identifier, SystemTemplate::TransportProtocols::J1939TpPg.requestable, SystemTemplate::TransportProtocols::J1939TpPg.pgn	
Mapping Rule	Mapping Type

This parameter can be derived from the Pdu and a combination of CanFrameTriggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00026]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmComUser/J1939RmComIPdu
BSW Parameter	BSW Type
J1939RmComIPduPGN	EcucIntegerParamDef
BSW Description	
PGN of the COM I-PDU.	
Template Description	
<p>ISignalIPdu: 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.</p> <p>A maximum of one dynamic length signal per IPdu is allowed.</p> <p>CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message.</p> <p>CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network.</p> <p>J1939TpPg.requestable: Parameter Group can be triggered by the J1939 request message.</p> <p>J1939TpPg.pgn: Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.j1939requestable, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier, SystemTemplate::TransportProtocols::J1939TpPg.requestable, SystemTemplate::TransportProtocols::J1939TpPg.pgn	
Mapping Rule	Mapping Type
This parameter can be derived fromSignalIPdu and a combination of CanFrameTriggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00033]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmDcmUser
BSW Parameter	BSW Type
J1939RmUserRequestPGN	EcucIntegerParamDef
BSW Description	

PGN of DMx PG supported by J1939Dcm.	
Template Description	
<p>Pdu: Collection of all Pdus that can be routed through a bus interface.</p> <p>CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message.</p> <p>CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network.</p> <p>J1939TpPg.requestable: Parameter Group can be triggered by the J1939 request message.</p> <p>J1939TpPg.pgn: Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.</p>	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.j1939requestable, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier, SystemTemplate::TransportProtocols::J1939TpPg.requestable, SystemTemplate::TransportProtocols::J1939TpPg.pgn	
Mapping Rule	Mapping Type
This parameter can be derived from the Pdu and a combination of CanFrameTriggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00070]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmRteUser
BSW Parameter	BSW Type
J1939RmUserRequestPGN	EcucIntegerParamDef
BSW Description	
PGN supported to be requested from this module. The PGNs supported by different modules should usually be disjunctive.	
Template Description	

Pdu: Collection of all Pdus that can be routed through a bus interface.	
CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message.	
CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network.	
J1939TpPg.requestable: Parameter Group can be triggered by the J1939 request message.	
J1939TpPg.pgn: Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.j1939requestable, SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier, SystemTemplate::TransportProtocols::J1939TpPg.requestable, SystemTemplate::TransportProtocols::J1939TpPg.pgn	
Mapping Rule	Mapping Type
This parameter can be derived from the Pdu and a combination of CanFrameTriggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00026]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmGeneral
BSW Parameter	BSW Type
J1939RmSupportRequest2	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling support of the Request2 PG. Please note: Transfer is not supported.	
Template Description	
Enables support for the Request2 PGN (RQST2).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.request2Support	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Rm_00073]

C.4 FlexRay

C.4.1 FlexRay Driver Mapping

BSW Module	BSW Context
Fr	Fr
BSW Parameter	BSW Type

FrGeneral	EcucParamConfContainerDef
BSW Description	
General configuration (parameters) of the FlexRay Driver module.	
Template Description	
FlexRay specific attributes to the physicalCluster	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster	
Mapping Rule	Mapping Type
Container shall be created if the ECU is connected to a FlexRay Cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00392]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrCtrlTestCount	EcucIntegerParamDef
BSW Description	
Maximum number of iterations the FlexRay controller hardware test is performed during controller initialization.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00001]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00393]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrDisableLPduSupport	EcucBooleanParamDef
BSW Description	

Enables or disabled API function Fr_DisableLPdu.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00455]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the Flexray driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Flexray driver will operate as an independent instance in each of the partitions.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00457]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrExtendedLPduReporting	EcucBooleanParamDef
BSW Description	
Enables or disables reporting of actual cycle and slot ID by Fr_TransmitTxLPdu, Fr_ReceiveRxLPdu, and Fr_CheckTxLPduStatus.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00459]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrIndex	EcucIntegerParamDef
BSW Description	
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00439]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrNumCtrlSupported	EcucIntegerParamDef
BSW Description	
Determines the maximum number of communication controllers that the driver supports.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00394]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrPrepareLPduSupport	EcucBooleanParamDef
BSW Description	
Enables or disables API function Fr_PrepareLPdu.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00453]

BSW Module	BSW Context
Fr	Fr/FrGeneral
BSW Parameter	BSW Type
FrReconfigLPduSupport	EcucBooleanParamDef
BSW Description	
Enables or disabled API function Fr_ReconfigLPdu.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00454]

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		BSW Type
FrRxStringentCheck		EcucBooleanParamDef
BSW Description		
If stringent check is enabled (true), received frames are accepted only if no slot status error occurred.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00002]

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		BSW Type
FrRxStringentLengthCheck		EcucBooleanParamDef
BSW Description		
If stringent check is enabled (true), received frames are accepted only if the received payload length matches the configured payload length.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00016]

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		BSW Type
FrVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables/disables the existence of the Fr_GetVersionInfo API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00396]

BSW Module	BSW Context	
Fr	Fr	
BSW Parameter	BSW Type	
FrMultipleConfiguration	EcucParamConfContainerDef	
BSW Description	This container contains the configuration parameters and sub containers of the AUTOSAR Fr module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Fr_00397]	

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration	
BSW Parameter	BSW Type	
FrController	EcucParamConfContainerDef	
BSW Description	Configuration of the individual controller.	
Template Description	FlexRay bus specific communication port attributes.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController	
Mapping Rule	Mapping Type	
Container shall be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Fr_00083]	

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrAbsoluteTimer	EcucParamConfContainerDef	
BSW Description	Specifies the absolute timer configuration parameters of the Fr.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Fr_00432]	

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrAbsoluteTimer	
BSW Parameter	BSW Type	

FrAbsTimerIdx	EcucIntegerParamDef
BSW Description	
Contains the index of an absolute timer contained in Fr on a certain FlexRay CC.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00433]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrControllerDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00452]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrControllerDemEventParameterRefs
BSW Parameter	BSW Type
FR_E_CTRL_TESTRESULT	EcucReferenceDef
BSW Description	
Reference to DEM event Id that is reported for FlexRay controller hardware test failure. If this parameter is not configured, no event reporting happens.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00005]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrCtrlEcucPartitionRef	EcucReferenceDef

BSW Description	
Maps one single Flexray controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Flexray driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00458]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrCtrlIdx	EcucIntegerParamDef
BSW Description	
Determines index of CC within Fr.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00400]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrFifo	EcucParamConfContainerDef
BSW Description	
One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.	
Template Description	
One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00009]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrAdmitWithoutMessageId	EcucBooleanParamDef
BSW Description	

Determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.	
Template Description	
Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.admitWithoutMessageld	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00006]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrBaseCycle	EcucIntegerParamDef
BSW Description	
FIFO cycle counter acceptance criteria.	
Template Description	
FIFO cycle counter acceptance criteria.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.baseCycle	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00007]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrChannels	EcucEnumerationParamDef
BSW Description	
FIFO channel admittance criteria.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.channel	
Mapping Rule	Mapping Type
If channelA is referenced set Parameter to FR_CHANNEL_A. If channelB is referenced set parameter to FR_CHANNEL_B. If two identical FlexrayFifoConfiguration elements exist with references to A and B only one FrFifo container shall be created (FR_CHANNEL_AB	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00449]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrCycleRepetition	EcucIntegerParamDef

BSW Description	
FIFO cycle counter acceptance criteria. Valid values are 1,2,4,5,8,10,16,20,32,40,50,64. Remark: Values 1,2,4,8,16,32,64 are valid only for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
FIFO cycle counter acceptance criteria.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.cycleRepetition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00008]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrFifoDepth	EcucIntegerParamDef
BSW Description	
FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.	
Template Description	
FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.fifoDepth	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00010]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrMsgIdMask	EcucIntegerParamDef
BSW Description	
FIFO message identifier acceptance criteria (Mask filter).	
Template Description	
FIFO message identifier acceptance criteria (Mask filter).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.msgIdMask	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00011]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrMsgIdMatch	EcucIntegerParamDef
BSW Description	
FIFO message identifier acceptance criteria (Match filter).	
Template Description	
FIFO message identifier acceptance criteria (Match filter).	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.msgIdMatch	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00012]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	BSW Type
FrRange	EcucParamConfContainerDef
BSW Description	
FIFO Frame Id range acceptance criteria.	
Template Description	
FIFO Frame Id range acceptance criteria.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange	
Mapping Rule	Mapping Type
create container for each Fifo configuration	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00013]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange
BSW Parameter	BSW Type
FrRangeMax	EcucIntegerParamDef
BSW Description	
Last Frameld of this range that will be accepted by the FIFO.	
Template Description	
Max Range.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange.rangeMax	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00014]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange
BSW Parameter	BSW Type
FrRangeMin	EcucIntegerParamDef
BSW Description	
First Frameld of this range that will be accepted by the FIFO.	
Template Description	
Min Range.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange.rangeMin	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_Fr_00015]
-------	-----------------

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPAllowHaltDueToClock	EcucBooleanParamDef	
BSW Description		
Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the CC is allowed to transition to POC:halt. If set to false, the CC will not transition to the POC:halt state but will enter or remain in the POC:normal passive state (self healing would still be possible)		
Template Description		
Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the Communication Controller is allowed to transition to POC:halt. If set to false, the Communication Controller will not transition to the POC:halt state but will enter or remain in the normal POC (passive State).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.allowHaltDueToClock		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00402]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPAllowPassiveToActive	EcucIntegerParamDef	
BSW Description		
Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to zero, the CC is not allowed to transition from POC:normal passive to POC:normal active		
Template Description		
Number of consecutive even/odd cycle pairs that shall have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to 0, the Communication Controller is not allowed to transition from POC:norm		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.allowPassiveToActive		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00403]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPChannels	EcucEnumerationParamDef	
BSW Description		

Channels to which the node is connected. Implementation Type: Fr_ChannelType	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.commConnector	
Mapping Rule	Mapping Type
If channelA refers the connector set parameter to FR_CHANNEL_A. If ChannelB refers the connector set parameter to FR_CHANNEL_B. If channelA and channelB refer the connector set parameter to FR_CHANNEL_AB,	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00404]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPClusterDriftDamping	EcucIntegerParamDef
BSW Description	
Local cluster drift damping factor used for rate correction [Microticks]. Remark: Upper limit 10 for FlexRay Protocol 3.0 compliance.	
Template Description	
The cluster drift damping factor used in clock synchronization rate correction in microticks	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.clusterDriftDamping	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00405]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPDecodingCorrection	EcucIntegerParamDef
BSW Description	
Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point [Microticks]. Remark: Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance. Upper limit 136 for FlexRay Protocol 3.0 compliance.	
Template Description	
Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.decodingCorrection	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00406]

BSW Module	BSW Context
-------------------	--------------------

Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		BSW Type
FrPDelayCompensationA		EcucIntegerParamDef
BSW Description		
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.		
Template Description		
Value used to compensate for reception delays on channel A Unit: Microticks. This optional parameter shall only be filled out if channel A is used.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.delayCompensationA		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00407]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		BSW Type
FrPDelayCompensationB		EcucIntegerParamDef
BSW Description		
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.		
Template Description		
Value used to compensate for reception delays on channel B. Unit: Microticks. This optional parameter shall only be filled out if channel B is used.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.delayCompensationB		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00408]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		BSW Type
FrPEternalSync		EcucBooleanParamDef
BSW Description		
Flag indicating whether the node is externally synchronized (operating as time gateway sink in an TT-E cluster) or locally synchronized. If FrPEternalSync is set to 'true' then FrPTwoKeySlotMode must also be set to 'true'. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		

Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TT-E Time Triggered External Sync cluster) or locally synchronized.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.external Sync	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00448]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPFallBackInternal	EcucBooleanParamDef
BSW Description	
Flag indicating whether a time gateway sink node will switch to local clock operation when synchronization with the time gateway source node is lost (FrPFallBackInternal = true) or will instead go to POC:ready (FrPFallBackInternal =false). Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.fallBack Internal	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00447]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPKeySlotId	EcucIntegerParamDef
BSW Description	
ID of the key slot, i.e., the slot used to transmit the startup frame, sync frame, or designated key slot frame. If this parameter is set to zero the node does not have a key slot. For Fr3.0: if the value is not provided in System Description it shall be configured to 0. For Fr2.1: if the value is not provided in System Description it is driver implementation specific which value to configure.	
Template Description	
ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedForStartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotI D	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_Fr_00411]
-------	-----------------

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPKeySlotOnlyEnabled	EcucBooleanParamDef
BSW Description	
Flag indicating whether or not the node shall enter key slot only mode following startup. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pSingleSlotEnabled.	
Template Description	
Flag indicating whether or not the node shall enter key slot only mode following startup.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotOnlyEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00425]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPKeySlotUsedForStartup	EcucBooleanParamDef
BSW Description	
Flag indicating whether the key slot is used to transmit a startup frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true.	
Template Description	
Flag indicating whether the Key Slot is used to transmit a startup frame.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotUsedForStartUp	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00412]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPKeySlotUsedForSync	EcucBooleanParamDef
BSW Description	
Flag indicating whether the key slot is used to transmit a sync frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true.	
Template Description	
Flag indicating whether the Key Slot is used to transmit a sync frame.	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotUsedForSync	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00413]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPLatestTx	EcucIntegerParamDef
BSW Description	
Number of the last minislot in which a frame transmission can start in the dynamic segment. Remark: Upper limit 7980 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
The number of the last minislot in which a transmission can start in the dynamic segment for the respective node	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.latestTX	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00414]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPMacroInitialOffsetA	EcucIntegerParamDef
BSW Description	
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].	
Template Description	
Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel A is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.macroInitialOffsetA	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00415]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPMacroInitialOffsetB	EcucIntegerParamDef
BSW Description	
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].	
Template Description	

Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel B is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.macroInitialOffsetB	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00416]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPMicroInitialOffsetA	EcucIntegerParamDef
BSW Description	
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationA and therefore it has to be set independently for each channel [Microticks].	
Template Description	
Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel A is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microInitialOffsetA	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00417]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPMicroInitialOffsetB	EcucIntegerParamDef
BSW Description	
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationB and therefore it has to be set independently for each channel [Microticks].	
Template Description	
Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel B is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microInitialOffsetB	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00418]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPMicroPerCycle	EcucIntegerParamDef	
BSW Description		
Nominal number of microticks in the communication cycle of the local node. If nodes have different microtick durations this number will differ from node to node [Microticks]. Remark: Lower limit 960 for FlexRay Protocol 3.0 compliance. Upper limit 640000 for FlexRay Protocol 2.1 Rev A compliance.		
Template Description		
The nominal number of microticks in a communication cycle		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microPerCycle		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Fr_00419]	

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPNmVectorEarlyUpdate	EcucBooleanParamDef	
BSW Description		
Flag indicating when the update of the Network Management Vector in the CHI shall take place. If FrPNmVectorEarlyUpdate is set to false, the update shall take place after the NIT. If FrPNmVectorEarlyUpdate is set to true, the update shall take place after the end of the static segment. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Flag indicating when the update of the Network Management Vector in the CHI shall take place. If set to false, the update shall take place after the NIT. If set to true, the update shall take place after the end of the static segment.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.nmVectorEarlyUpdate		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Fr_00444]	

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPOffsetCorrectionOut	EcucIntegerParamDef	
BSW Description		

Magnitude of the maximum permissible offset correction value [Microticks]. Remark: Upper limit 15567 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 15 for FlexRay Protocol 3.0 compliance.	
Template Description	
Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.offsetCorrectionOut	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00421]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPOffsetCorrectionStart	EcucIntegerParamDef
BSW Description	
Start of the offset correction phase within the NIT, expressed as the number of macroticks from the start of cycle [Macroticks]. Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gOffsetCorrectionStart. Remark: Lower limit 9 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Start of the offset correction phase within the Network Idle Time (NIT), expressed as the number of macroticks from the start of cycle. Unit: macroticks	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.offsetCorrectionStart	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00450]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPPayloadLengthDynMax	EcucIntegerParamDef
BSW Description	
Maximum payload length for dynamic frames [16 bit words].	
Template Description	
Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.maximumDynamicPayloadLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00422]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController

BSW Parameter		BSW Type	
FrPRateCorrectionOut		EcucIntegerParamDef	
BSW Description			
Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle [Microticks]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift. Lower limit 3 for FlexRay Protocol 3.0 compliance. Upper limit 1923 for FlexRay Protocol 2.1 Rev A compliance.			
Template Description			
Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrectionOut) Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.rateCorrectionOut			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Fr_00423]	

BSW Module		BSW Context	
Fr		Fr/FrMultipleConfiguration/FrController	
BSW Parameter		BSW Type	
FrPSamplesPerMicrotick		EcucEnumerationParamDef	
BSW Description			
Number of samples per microtick. Remark: Allowed range N1SAMPLES, N2SAMPLES for FlexRay Protocol 3.0 compliance.			
Template Description			
Number of samples per microtick			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.samplesPerMicrotick			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Fr_00424]	

BSW Module		BSW Context	
Fr		Fr/FrMultipleConfiguration/FrController	
BSW Parameter		BSW Type	
FrPSecondKeySlotId		EcucIntegerParamDef	
BSW Description			
ID of the second key slot, in which a second startup frame shall be sent when operating as a coldstart node in a TT-L or TT-D cluster. If this parameter is set to zero the node does not have a second key slot. Remark: Set to 0 for FlexRay Protocol 2.1 Rev A compliance.			
Template Description			

ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.secondKeySlotId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00445]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPTwoKeySlotMode	EcucBooleanParamDef
BSW Description	
Flag indicating whether node operates as a coldstart node in a TT-E or TT-L cluster. If pTwoKeySlotMode is set to true then both pKeySlotUsedForSync and pKeySlotUsedForStartup must also be set to true. If pExternalSync is set to true then pTwoKeySlotMode must also be set to true. Remark: Set to false for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.twoKeySlotMode	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00446]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPWakeupChannel	EcucEnumerationParamDef
BSW Description	
Channel used by the node to send a wakeup pattern. FrPWakeupChannel must be selected from among the channels configured by FrPChannels.	
Template Description	
Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.wakeUpChannel	
Mapping Rule	Mapping Type
If channelA refers to the FlexrayCommunicationConnector and wakeUpChannel=true then FrPWakeupChannel = FR_CHANNEL_A. If channelB refers to the FlexrayCommunicationConnector and wakeUpChannel = true then FrPWakeupChannel = FR_CHANNEL_B.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00426]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPWakeupPattern	EcucIntegerParamDef	
BSW Description		
<p>Number of repetitions of the wakeup symbol that are combined to form a wakeup pattern when the node enters the POC:wakeup send state. Remark: Lower limit 2 for FlexRay Protocol 2.1 Rev A compliance.</p>		
Template Description		
<p>Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.wakeUpPattern		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00427]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPdAcceptedStartupRange	EcucIntegerParamDef	
BSW Description		
<p>Expanded range of measured clock deviation allowed for startup frames during integration [Microticks]. Remark: Upper limit 1875 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 29 for FlexRay Protocol 3.0 compliance.</p>		
Template Description		
<p>Expanded range of measured clock deviation allowed for startup frames during integration. Unit: microtick</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.acceptedStartupRange		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Fr_00428]

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter	BSW Type	
FrPdListenTimeout	EcucIntegerParamDef	
BSW Description		
<p>Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster [Microticks]. Remark: Lower limit 1926 for FlexRay Protocol 3.0 compliance. Upper limit 1283846 for FlexRay Protocol 2.1 Rev. A compliance.</p>		
Template Description		
<p>Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks</p>		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.listenTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00429]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	BSW Type
FrPdMicrotick	EcucEnumerationParamDef
BSW Description	
Duration of a microtick. Remark: Allowed range T12_5NS, T25NS, T50NS for FlexRay Protocol 3.0 compliance.	
Template Description	
Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microtickDuration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00431]

C.4.2 FlexRay Interface Mapping

BSW Module	BSW Context
Frlf	Frlf
BSW Parameter	BSW Type
FrlfConfig	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR Frlf module.	
Template Description	
FlexRay specific attributes to the physicalCluster	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster	
Mapping Rule	Mapping Type
Container shall be created if the ECU is connected to a FlexRay Cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06001]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig
BSW Parameter	BSW Type
FrlfCluster	EcucParamConfContainerDef
BSW Description	

This container specifies a Frlf Cluster and all related data which is required to enable communication of the Cluster. A Cluster may consist of more than one Controller.	
Template Description	
FlexRay specific attributes to the physicalCluster	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster	
Mapping Rule	Mapping Type
Container shall be created if the ECU is connected to a FlexRay Cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05366]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfClstIdx	EcucIntegerParamDef
BSW Description	
This parameter provides a zero-based consecutive index of the FlexRay Clusters. Upper layer BSW modules and the Frlf itself use this index to identify a FlexRay Cluster.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06002]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfClusterDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06091]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_ACS_CH_A	EcucReferenceDef
BSW Description	

Reference to the DemEventParameter which shall be issued when an error in ACS on channel A was detected. If the reference is not configured the error shall not be reported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06097]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_ACS_CH_B	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in ACS on channel B was detected. If the reference is not configured the error shall not be reported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06098]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_NIT_CH_A	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in NIT on channel A was detected. If the reference is not configured the error shall not be reported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06093]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_NIT_CH_B	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in NIT on channel B was detected. If the reference is not configured the error shall not be reported.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06094]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_SW_CH_A	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in SW on channel A was detected. If the reference is not configured the error shall not be reported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06095]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfClusterDemEventParameterRefs
BSW Parameter	BSW Type
FRIF_E_SW_CH_B	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in SW on channel B was detected. If the reference is not configured the error shall not be reported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06096]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfController	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of FlexRay CC.	
Template Description	
FlexRay bus specific communication port attributes.	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController	
Mapping Rule	Mapping Type
Container shall be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05363]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController
BSW Parameter	BSW Type
FrlfCtrlIdx	EcucIntegerParamDef
BSW Description	
This parameter provides a zero-based consecutive index of the FlexRay Communication Controllers. Upper layer BSW modules and the Frlf itself use this index to identify a FlexRay CC.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06045]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController
BSW Parameter	BSW Type
FrlfFrCtrlRef	EcucReferenceDef
BSW Description	
Reference to a Controller, which is handled by a specific Driver. This reference is unique for the ECU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06044]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController
BSW Parameter	BSW Type
FrlfFrameTriggering	EcucParamConfContainerDef
BSW Description	
A Frame triggering contains the communication parameters of the FlexRay Frame as well as a reference to the Frame Construction Plan.	
Template Description	
FlexRay specific attributes to the FrameTriggering	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering	

Mapping Rule		Mapping Type
If a FlexrayFrameTriggering exists in the System Extract that is connected via a FramePort reference to the regarded Ecu the following two cases exist: 1) If the FlexrayFrameTriggering contains exactly one FlexrayAbsolutelyScheduledTiming then only one FrlfFrameTriggering container shall be created. 2) If the FlexrayFrameTriggering contains more than one FlexrayAbsolutelyScheduledTiming (e.g. to describe a multiple sending within one communication cycle) this FrlfFrameTriggering container shall be created once per defined FlexrayAbsolutelyScheduledTiming. Each created FrlfFrameTriggering container shall refer to the same FrlfFrameStructure.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06090]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter		BSW Type
FrlfAllowDynamicLSduLength		EcucBooleanParamDef
BSW Description		
Allows L-PDU length reduction ('FrlfLSduLength' defines max. length) and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU.		
Template Description		
Allows L-PDU length reduction and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU.		
If this attribute is set to true than the referenced Frame length attribute defines the max. length.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.allowDynamicLSduLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06049]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter		BSW Type
FrlfAlwaysTransmit		EcucBooleanParamDef
BSW Description		
Defines whether the driver's API function Fr_TransmitTxLPdu() shall always be called for this L-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_00013]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	

BSW Parameter		BSW Type
FrlfBaseCycle		EcucIntegerParamDef
BSW Description		
This parameter contains the FlexRay Base Cycle used to transmit this FlexRay Frame.		
Template Description		
The first communication cycle where the frame is sent.		
This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.BaseCycle		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06051]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter		BSW Type
FrlfChannel		EcucEnumerationParamDef
BSW Description		
This parameter contains the FlexRay Channel used to transmit this FlexRay Frame.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.frameTriggering		
Mapping Rule		Mapping Type
FrameTriggering element in the System Template is aggregated by the Physical Channel that is used to transmit this FlexRay Frame		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06052]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter		BSW Type
FrlfCycleRepetition		EcucIntegerParamDef
BSW Description		
This parameter contains the FlexRay Cycle Repetition used to transmit this FlexRay Frame.		
Possible values for FlexRay Protocol version 2.1: 1,2,4,8,16,32,64		
Possible values for FlexRay Protocol version 3.0: 1,2,4,5,8,10,16,20,32,40,50,64		
Template Description		
The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.CycleRepetition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06053]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfFrameStructureRef	EcucReferenceDef	
BSW Description	Reference to the Construction Plan of the FlexRay Frame.	
Template Description		
M2 Parameter	SystemTemplate::Fibex::FibexCore::CoreCommunication::FrameTriggering.frame	
Mapping Rule	Mapping Type	
Reference shall comply to the reference in the System Description between the FrameTriggering element and the Frame.element.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06048]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfFrameTriggeringDemEventParameterRefs	EcucParamConfContainerDef	
BSW Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06099]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering/FrlfFrameTriggeringDemEventParameterRefs	
BSW Parameter	BSW Type	
FRIF_E_LPDU_SLOTSTATUS	EcucReferenceDef	
BSW Description	Reference to DEM event Id that is reported when FlexRay driver module detects slot errors. If this parameter is not configured, no event reporting happens.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_00009]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfLsduLength	EcucIntegerParamDef	
BSW Description		
The payload length of the Frame is given here. This parameter is required for validation if configured PDUs and update information fits into the Frame at configuration time [bytes].		
Template Description		
The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay).		
The frameLength of zero bytes is allowed.		
Please consider also TPS_SYST_02255.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame.frameLength		
Mapping Rule		Mapping Type
Find Frame that is referenced by the regarded FrameTriggering and use the frameLength attribute		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06054]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfMessageld	EcucIntegerParamDef	
BSW Description		
The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.		
Template Description		
The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.messageId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_00010]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfPayloadPreamble	EcucBooleanParamDef	
BSW Description		
Switching the Payload Preamble bit.		
Template Description		
Switching the Payload Preamble bit.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.payloadPreambleIndicator		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06055]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering	
BSW Parameter	BSW Type	
FrlfSlotId	EcucIntegerParamDef	
BSW Description		
This parameter contains the FlexRay Slot ID used to transmit this FlexRay Frame.		
Template Description		
<p>In the static part the SlotID defines the slot in which the frame is transmitted. The SlotID also determines, in combination with FlexrayCluster::numberOfStaticSlots, whether the frame is sent in static or dynamic segment.</p> <p>In the dynamic part, the slot id is equivalent to a priority. Lower dynamic slot ids are all sent until the end of the dynamic segment. Higher numbers, which were ignored that time, have to wait one cycle and then shall try again.</p> <p>minValue: 1</p> <p>maxValue: 2047</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayAbsolutelyScheduledTiming.slotID		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06056]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController	
BSW Parameter	BSW Type	
FrlfLPdu	EcucParamConfContainerDef	
BSW Description		
Reference to a L-PDU index		
Template Description		
Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame		
Mapping Rule		Mapping Type
Create container for each FlexRay Frame that is transmitted or received via the regarded communication controller..		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_05364]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfLPdu	
BSW Parameter	BSW Type	
FrlfLPduldx	EcucIntegerParamDef	

BSW Description	
This parameter identifies the L-PDU in the interaction between FlexRay Interface and FlexRay Driver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06058]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfLPdu
BSW Parameter	BSW Type
FrlfReconfigurable	EcucBooleanParamDef
BSW Description	
This parameter specifies that this LPdu is reconfigurable using Frlf_ReconfigLPdu. This means that this LPdu can be assigned to a different FrameTriggering at runtime. However, this reconfiguration is limited by hardware constraints. The direction of the LPdu cannot be reconfigured.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00008]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfLPdu
BSW Parameter	BSW Type
FrlfVBTriggeringRef	EcucReferenceDef
BSW Description	
Reference to the assigned Frame triggering.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06057]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController
BSW Parameter	BSW Type
FrlfTransceiver	EcucParamConfContainerDef
BSW Description	
Up to two FlexRay Transceivers may connect a Controller to a Cluster. This container realizes a Controller-Transceiver assignment.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05391]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfTransceiver
BSW Parameter	BSW Type
FrlfClusterChannel	EcucEnumerationParamDef
BSW Description	
This parameter identifies to which one of the two Channels (A, B, A and B) of the Cluster the Transceiver is connected. FrlfClusterChannel shall map to Fr_ChannelType: FRIF_CHANNEL_A == FR_CHANNEL_A FRIF_CHANNEL_B == FR_CHANNEL_B FR_CHANNEL_AB shall not be used.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06062]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfTransceiver
BSW Parameter	BSW Type
FrlfFrTrcvChannelRef	EcucReferenceDef
BSW Description	
Reference to a Transceiver Driver Channel. This reference is unique for the ECU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06061]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfDetectNITError	EcucBooleanParamDef
BSW Description	
Indicates whether NIT error status of each cluster shall be detected or not.	
Template Description	
Indicates whether NIT error status of each cluster shall be detected or not.	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.detectNitError	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00003]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGChannels	EcucEnumerationParamDef
BSW Description	
The channels that are used by the cluster.	
Implementation Type: Fr_ChannelType	
Template Description	
FlexRay specific attributes to the physicalChannel	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayPhysicalChannel	
Mapping Rule	Mapping Type
The channels that are used by the cluster are described in the System Template by the CommunicationCluster-PhysicalChannel relationship.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06006]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGColdStartAttempts	EcucIntegerParamDef
BSW Description	
Maximum number of times a node in the cluster is permitted to attempt to start the cluster by initiating schedule synchronization	
Template Description	
The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.coldStartAttempts	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06008]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGCycleCountMax	EcucIntegerParamDef
BSW Description	
Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.cycleCountMax	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06086]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGListenNoise	EcucIntegerParamDef
BSW Description	
Upper limit for the start up listen timeout and wake up listen timeout in the presence of noise. It is used as a multiplier of the node parameter pdListenTimeout.	
Template Description	
Upper limit for the start up and wake up listen timeout in the presence of noise. Expressed as a multiple of the cluster constant pdListenTimeout. Unit microticks	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.listenNoise	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06009]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGMacroPerCycle	EcucIntegerParamDef
BSW Description	
Number of macroticks in a communication cycle.	
Note: Lower limit 10 for FlexRay Protocol 2.1 Rev. A compliance	
Template Description	
The number of macroticks in a communication cycle	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.macroPerCycle	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06010]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGMaxWithoutClockCorrectFatal	EcucIntegerParamDef
BSW Description	
Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state. [Even/odd cycle pairs].	

Template Description	
Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.maxWithoutClockCorrectionFatal	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06011]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGMaxWithoutClockCorrectPassive	EcucIntegerParamDef
BSW Description	
Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state. [Even/Odd cycle pairs]	
Template Description	
Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.maxWithoutClockCorrectionPassive	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06012]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGNetworkManagementVectorLength	EcucIntegerParamDef
BSW Description	
Length of the Network Management vector in a cluster [bytes]	
Template Description	
Length of the Network Management vector in a cluster [bytes]	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.networkManagementVectorLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06013]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type

FrlfGNumberOfMinislots	EcucIntegerParamDef
BSW Description	
Number of minislots in the dynamic segment	
Remark: Upper limit 7986 for FlexRay Protocol 2.1 Rev. A compliance	
Template Description	
Number of Minislots in the dynamic segment.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.numberOfMinislots	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06014]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGNumberOfStaticSlots	EcucIntegerParamDef
BSW Description	
Number of static slots in the static segment	
Template Description	
The number of static slots in the static segment.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.numberOfStaticSlots	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06015]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGPayloadLengthStatic	EcucIntegerParamDef
BSW Description	
Payload length of a static frame [16 bit words]	
Template Description	
Globally configured payload length of a static frame. Unit: 16-bit WORDS.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.payloadLengthStatic	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06018]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGSyncFrameIDCountMax	EcucIntegerParamDef
BSW Description	

Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.	
Template Description	
Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.syncFrameIdCountMax	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06019]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdActionPointOffset		EcucIntegerParamDef
BSW Description		
Number of macroticks the action point is offset from the beginning of a static slot.		
Template Description		
The offset of the action point in networks		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.actionPointOffset		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06020]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdBit		EcucEnumerationParamDef
BSW Description		
Nominal bit time in seconds		
Template Description		
Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPerBit * gdSampleClockPeriod. Unit: seconds (gdBit)		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.bit		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06021]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdCasRxLowMax		EcucIntegerParamDef
BSW Description		
Upper limit of the CAS acceptance windows [gdBit]		
Remark: Range 67 to 99 for FlexRay Protocol 2.1 Rev. A compliance		

Template Description	
Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.casRxLowMax	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06024]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdCycle		EcucFloatParamDef
BSW Description		
Length of the cycle, expressed in [s] Remark: Lower limit 0.000024 for FlexRay Protocol 3.0 compliance.		
Template Description		
Length of the cycle. Unit: seconds		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.cycle		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06025]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdDynamicSlotIdlePhase		EcucIntegerParamDef
BSW Description		
Duration of the idle phase within a dynamic slot [Minislots].		
Template Description		
The duration of the dynamic slot idle phase in minislots.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.dynamicSlotIdlePhase		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06026]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type
FrlfGdIgnoreAfterTx		EcucIntegerParamDef
BSW Description		
Duration for which the bitstrobing is paused after transmission [gdBit].		
Remark: Set to 0 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Duration for which the bitstrobing is paused after transmission [gdBit].		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.ignoreAfterTx	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00012]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGdMacrotick	EcucFloatParamDef
BSW Description	
Duration of the cluster wide nominal macrotick, expressed in s	
Template Description	
Duration of the cluster wide nominal macrotick, expressed in s.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.macrotickDuration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06027]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGdMiniSlotActionPointOffset	EcucIntegerParamDef
BSW Description	
Number of Macroticks the Minislot action point is offset from the beginning of a Minislot [Macroticks].	
Template Description	
The Offset of the action point within a minislot. Unit: macroticks	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.minislotActionPointOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06032]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGdMinislot	EcucIntegerParamDef
BSW Description	
Duration of a minislot [Macroticks]	
Template Description	
The duration of a minislot (dynamic segment). Unit: macroticks.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.minislotDuration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_Frlf_06033]
-------	-------------------

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdNit	EcucIntegerParamDef	
BSW Description	Duration of the Network Idle Time [Macroticks]	
Remark: Upper limit 805 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description	The duration of the network idle time in macroticks	
M2 Parameter	SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.networkIdleTime	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06034]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdSampleClockPeriod	EcucEnumerationParamDef	
BSW Description	Sample clock period	
Template Description	Sample clock period. Unit: seconds	
M2 Parameter	SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.sampleClockPeriod	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06035]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdStaticSlot	EcucIntegerParamDef	
BSW Description	Duration of a static slot [Macroticks]. Remark: Range 4-661 for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	The duration of a slot in the static segment. Unit: macroticks	
M2 Parameter	SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.staticSlotDuration	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06036]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type	
FrlfGdSymbolWindow		EcucIntegerParamDef	
BSW Description			
Duration of the symbol window [Macroticks].			
Remark: Range 0-142 for FlexRay Protocol 2.1 Rev. A compliance.			
Template Description			
Number of bits in the Transmission Start Sequence [gdBits].			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.transmissionStartSequenceDuration			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06037]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type	
FrlfGdSymbolWindowActionPointOffset		EcucIntegerParamDef	
BSW Description			
Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].			
Remark: Set to GdActionPointOffset for FlexRay Protocol 2.1 Rev. A compliance.			
Template Description			
Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.symbolWindowActionPointOffset			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_00011]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type	
FrlfGdTSSTransmitter		EcucIntegerParamDef	
BSW Description			
Number of bits in the Transmission Start Sequence [gdBits].			
Remark: Lower limit 3 for FlexRay Protocol 2.1 Rev. A compliance.			
Template Description			
Number of bits in the Transmission Start Sequence [gdBits].			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.transmissionStartSequenceDuration			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06038]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdWakeupRxIdle	EcucIntegerParamDef	
BSW Description		
Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle. Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit:bitDuration		
Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxIdle		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06039]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdWakeupRxLow	EcucIntegerParamDef	
BSW Description		
Number of bits used by the node to test the duration of the LOW phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow. Lower limit 11 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit:bitDuration		
Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxLow		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06040]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter	BSW Type	
FrlfGdWakeupRxWindow	EcucIntegerParamDef	
BSW Description		
The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow. Upper limit 301 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		

The size of the window used to detect wakeups [gdBit].	
Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxWindow	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06041]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGdWakeupTxActive	EcucIntegerParamDef
BSW Description	
Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP [gdBit].	
Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxLow.	
Template Description	
Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP. Unit:bitDuration	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupTxActive	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06043]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfGdWakeupTxIdle	EcucIntegerParamDef
BSW Description	
Number of bits used by the node to transmit the 'idle' part of a wakeup symbol [gdBit].	
Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxIdle.	
Template Description	
Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gDbit	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupTxIdle	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06042]

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		BSW Type	
FrlfJobList		EcucParamConfContainerDef	
BSW Description			
This container specifies a list of all FlexRay Jobs of the Cluster to be performed by Frlf_JobListExec_<FrlfCluster.ShortName>().			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_05367]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster/FrlfJobList	
BSW Parameter		BSW Type	
FrlfAbsTimerRef		EcucReferenceDef	
BSW Description			
Reference to the absolute timer to be used to trigger the interrupt whose ISR contains the Frlf_JobListExec_<FrlfCluster.ShortName>() function.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06063]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster/FrlfJobList	
BSW Parameter		BSW Type	
FrlfJob		EcucParamConfContainerDef	
BSW Description			
A job may contain more than one operation that are executed at a specific point in time.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_05368]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob	
BSW Parameter		BSW Type	

FrlfCommunicationOperation	EcucParamConfContainerDef
BSW Description	
A separate operation which is part of a FlexRay Job and defines what type of action is executed.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05369]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob/FrlfCommunicationOperation
BSW Parameter	BSW Type
FrlfCommunicationAction	EcucEnumerationParamDef
BSW Description	
The action to be performed in the FlexRay Operation	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06067]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob/FrlfCommunicationOperation
BSW Parameter	BSW Type
FrlfCommunicationOperationIdx	EcucIntegerParamDef
BSW Description	
For each FlexRay Communication Job, this index spans a range of zero-based consecutive values and thus defines the order of the FlexRay Communication Operation in the respective FlexRay Communication Job.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06068]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob/FrlfCommunicationOperation
BSW Parameter	BSW Type
FrlfLPdulIdxRef	EcucReferenceDef
BSW Description	
Reference to a L-PDU index	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06066]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob/FrlfCommunicationOperation
BSW Parameter	BSW Type
FrlfRxComOpMaxLoop	EcucIntegerParamDef
BSW Description	
Defines the maximum number of loops for the receive RECEIVE_AND_INDICATE (Use case: emptying a FIFO). Please note that the parameter is mandatory if FrlfCommunicationAction parameter is set to RECEIVE_AND_INDICATE. For all other operations this parameter can be ignored.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00007]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob
BSW Parameter	BSW Type
FrlfCycle	EcucIntegerParamDef
BSW Description	
The FlexRay Cycle in which the communication operation will execute this job	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06064]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob
BSW Parameter	BSW Type
FrlfMacrotick	EcucIntegerParamDef
BSW Description	
Macrotick offset in the Cycle [Macrotick]	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06065]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfJobList/FrlfJob
BSW Parameter	BSW Type
FrlfMaxIsrDelay	EcucIntegerParamDef
BSW Description	
The maximum delay in macroticks the Frlf_JobListExec_<FrlfCluster.ShortName>() function is processed after the absolute timer interrupt was triggered.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06004]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfMainFunctionPeriod	EcucFloatParamDef
BSW Description	
The execution cycle of the Frlf_MainFunction_<FrlfCluster.ShortName>() in seconds. The Frlf does not require this information but the BSW scheduler, which invokes the cluster main functions, needs it in order to plan its tasks.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06003]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	BSW Type
FrlfSafetyMargin	EcucIntegerParamDef
BSW Description	
Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has been resynchronized.	
Template Description	

Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has been resynchronized.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.safetyMargin	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00004]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig
BSW Parameter	BSW Type
FrlfFrameStructure	EcucParamConfContainerDef
BSW Description	
The Frame structure specifies a Construction Plan how a Frame is assembled with PDUs and their respective Update-Bits.	
Template Description	
Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame	
Mapping Rule	Mapping Type
Create container for each FlexRay Frame that is transmitted or received by the regarded ECU. IPduToFrameMapping element in the System Template contains the construction plan.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05370]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfFrameStructure
BSW Parameter	BSW Type
FrlfByteOrder	EcucEnumerationParamDef
BSW Description	
This parameter defines the ByteOrder of all Pdus that are mapped into the Frame.	
The absolute position of a Pdu in the Frame is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the FrlfPduOffset indicates the position of the most significant bit in the Frame. If LITTLE_ENDIAN is specified, the FrlfPduOffset indicates the position of the least significant bit in the Frame.	
Template Description	
This attribute defines the order of the bytes of the Pdu and the packing into the Frame. Please consider that [constr_3246] and [constr_3222] are restricting the usage of this attribute.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.packingByteOrder	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06113]

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfFrameStructure	
BSW Parameter		BSW Type	
FrlfPduInFrame		EcucParamConfContainerDef	
BSW Description			
This container holds all the information about a PDU in a FlexRay Frame.			
Template Description			
A PduToFrameMapping defines the composition of Pdus in each frame.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping			
Mapping Rule		Mapping Type	
Container shall be created for each IPduToFrameMapping element inside the frame.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_05371]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfFrameStructure/FrlfPduInFrame	
BSW Parameter		BSW Type	
FrlfPduOffset		EcucIntegerParamDef	
BSW Description			
The value specifies the offset of the PDU within the Frame [bytes].			
Template Description			
This attribute describes the bitposition of a Pdu within a Frame.			
<p>Please note that the absolute position of the Pdu in the Frame is determined by the definition of the packingByteOrder attribute. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the Frame. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the Frame. 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>The Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed.</p>			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.startPosition			
Mapping Rule		Mapping Type	
Please note that the startPosition attribute is defined in bits and the FrlfPdu Offset parameter is defined in bytes.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06070]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfFrameStructure/FrlfPduInFrame	
BSW Parameter		BSW Type	
FrlfPduRef		EcucReferenceDef	
BSW Description			
This is the reference to the local definition of a PDU.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06069]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfFrameStructure/FrlfPduInFrame
BSW Parameter	BSW Type
FrlfPduUpdateBitOffset	EcucIntegerParamDef
BSW Description	
This value specifies where the PDU's Update-Bit is stored in the Frame (bit location of PDU's Update-Bit in the FlexRay Frame).	
Template Description	
Indication to the receivers that the corresponding Pdu was updated by the sender. This attribute describes the position of the update bit in the frame that aggregates this PDU-toFrameMapping. Length is always one bit.	
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 Frame 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.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.updateIndicationBitPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06071]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig
BSW Parameter	BSW Type
FrlfMaxPduCnt	EcucIntegerParamDef
BSW Description	
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06121]

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig	
BSW Parameter		BSW Type	
FrlfPdu		EcucParamConfContainerDef	
BSW Description			
Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.			
Template Description			
Collection of all Pdus that can be routed through a bus interface.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu			
Mapping Rule			Mapping Type
The container shall be created for each Pdu that is contained in a FlexRay Frame.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_05372]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfPdu	
BSW Parameter		BSW Type	
FrlfPduDirection		EcucChoiceContainerDef	
BSW Description			
A PDU is either transmit or receive			
Template Description			
Communication Direction of the Connector Port (input or output Port).			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreTopology::CommConnectorPort.communicationDirection			
Mapping Rule			Mapping Type
The PduTriggering contains a reference to a IPduPort with the communication Direction.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06072]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection	
BSW Parameter		BSW Type	
FrlfRxPdu		EcucParamConfContainerDef	
BSW Description			
Receive PDU			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_05373]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfRxPdu	
BSW Parameter		BSW Type	

FrlfRxIndicationName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter FrlfUserRxIndicationUL. If FrlfUserRxIndicationUL equals FR_TP, FR_AR_TP, FR_NM, PDUR, FR_TSYN or XCP, the name of the <User_RxIndication> is fixed. If FrlfUserRxIndicationUL equals CDD, the name of the <User_RxIndication> is selectable.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00016]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfRxPdu
BSW Parameter	BSW Type
FrlfRxPduRef	EcucReferenceDef
BSW Description	
Reference to the external PDU definition.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06073]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfRxPdu
BSW Parameter	BSW Type
FrlfUserRxIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the indication of the successfully received FrlfRxPdu has to be routed via <User_RxIndication>. This <User_RxIndication> has to be invoked when the indication of the configured FrlfRxPdu will be received by a Rx indication event from the FR Driver module. If no upper layer (UL) module is configured, no <User_RxIndication> has to be called in case of a Rx indication event of the FrlfRxPdu from the FR Driver module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00017]

BSW Module	BSW Context
-------------------	--------------------

Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection	
BSW Parameter		BSW Type
FrlfTxPdu	EcucParamConfContainerDef	
BSW Description		
This container specifies transmission PDUs.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_05374]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu	
BSW Parameter		BSW Type
FrlfConfirm	EcucBooleanParamDef	
BSW Description		
Defines whether the transmission of a PDU should be checked and confirmed to the PDU owning BSW module. If "FrlfUserTxUL" is configured as FR_TSYN then this parameter has to be set to FALSE for this PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06075]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu	
BSW Parameter		BSW Type
FrlfCounterLimit	EcucIntegerParamDef	
BSW Description		
This value states the maximum number of indication of ready PDU data to the Frlf (i.e. maximum number of invocations of Frlf_Transmit) without an intermediate transmission of the PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Frlf_06076]

BSW Module	BSW Context	
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu	
BSW Parameter		BSW Type

FrlfImmediate	EcucBooleanParamDef
BSW Description	
Defines whether the PDU is transmitted immediate or decoupled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06077]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfNoneMode	EcucBooleanParamDef
BSW Description	
Using the "None-Mode" which means that there is no API Frlf_Transmit call of the upper layer for this PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06050]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfTxConfirmationName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter FrlfUserTxUL. If FrlfUserTxUL equals FR_TP, FR_AR_TP, FR_NM, PDUR or XCP, the name of the <User_TxConfirmation> is fixed. If FrlfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00014]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfTxPduld	EcucIntegerParamDef

BSW Description	
The global PDU identifier, which has to be used by the upper layer BSW module. The identifier has to be zero based and consecutive.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06078]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfTxPduRef	EcucReferenceDef
BSW Description	
Reference to the external PDU definition.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06074]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfUserTriggerTransmitName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter FrlfUserTxUL. If FrlfUserTxUL equals FR_TP, FR_AR_TP, FR_NM, PDUR, FR_TSYN or XCP the name of the <User_TriggerTransmit> is fixed. If FrlfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06084]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfPdu/FrlfPduDirection/FrlfTxPdu
BSW Parameter	BSW Type
FrlfUserTxUL	EcucEnumerationParamDef
BSW Description	

This parameter defines the upper layer (UL) module to which the trigger of the Pdu to be transmitted (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted Pdu has to be routed (via the <User_TxConfirmation>). Please note that handle IDs which are used in callback functions are defined by the upper layer module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00015]

BSW Module	BSW Context
Frlf	Frlf
BSW Parameter	BSW Type
FrlfGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the FlexRay Interface.	
Template Description	
FlexRay specific attributes to the physicalCluster	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster	
Mapping Rule	Mapping Type
Container shall be created if the ECU is connected to a FlexRay Cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_05360]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfAbsTimerIdx	EcucIntegerParamDef
BSW Description	
Maximum number of supported absolute timers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06112]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfAllSlotsSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable/disable of switching from key-slot / single-slot mode to all slot mode.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06108]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfBusMirroringSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable/disable reporting received/transmitted frames to the Bus Mirroring module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06124]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfCancelTransmitSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to request the cancellation of the I-PDU transmission to FrDrv.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00002]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06080]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfDisableLPduSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to disables the hardware resource of a LPdu for transmission/reception.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06110]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfDisableTransceiverBranchSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to disable branches of an active star.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06102]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfEnableTransceiverBranchSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable branches of an active star.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06103]

BSW Module	BSW Context	
Frlf	Frlf/FrlfGeneral	
BSW Parameter		BSW Type
FrlfFreeOpAApiName		EcucStringParamDef
BSW Description		
API name that is called when FREE_OP_A is selected as communication operation. See also chapter 8.8.3 Configurable Interfaces.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06118]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfGeneral	
BSW Parameter		BSW Type
FrlfFreeOpBApiName		EcucStringParamDef
BSW Description		
API name that is called when FREE_OP_B is selected as communication operation. See also chapter 8.8.3 Configurable Interfaces.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06119]	

BSW Module	BSW Context	
Frlf	Frlf/FrlfGeneral	
BSW Parameter		BSW Type
FrlfFreeOpsHeader		EcucStringParamDef
BSW Description		
Defines header file for configurable FREE_OP_A / FREE_OP_B functions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Frlf_06120]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfGeneral	
BSW Parameter		BSW Type	
FrlfGetClockCorrectionSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable Frlf support to enable/disable of polling the FlexRay Driver to getting CC clock correction values.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06106]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfGeneral	
BSW Parameter		BSW Type	
FrlfGetGetChannelStatusSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable Frlf support to enable/disable of polling the FlexRay Driver to getting error information about the FlexRay communications bus.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06105]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfGeneral	
BSW Parameter		BSW Type	
FrlfGetNmVectorSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable Frlf support to request the FlexRay hardware NMVector.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Frlf_06114]	

BSW Module		BSW Context	
Frlf		Frlf/FrlfGeneral	
BSW Parameter		BSW Type	

FrlfGetNumOfStartupFramesSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable/disable of polling the FlexRay Driver for the actual number of received startup frames on the bus.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06104]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfGetSyncFrameListSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable/disable of polling the FlexRay Driver to getting a list of actual received sync frames.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06107]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfGetTransceiverErrorSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to get the FlexRay Transceiver errors by calling the FlexRay Transceiver module.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06101]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfGetWakeupRxStatusSupport	EcucBooleanParamDef
BSW Description	

Configuration parameter to enable/disable Frlf support to get the wakeup received information from the FlexRay controller.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06111]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfNumClstSupported	EcucIntegerParamDef
BSW Description	
Maximum number of FlexRay Clusters that the FlexRay Interface supports.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06081]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfNumCtrlSupported	EcucIntegerParamDef
BSW Description	
Maximum number of FlexRay CCs that the FlexRay Interface supports	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06082]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfPublicCddHeaderFile	EcucStringParamDef
BSW Description	
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06116]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfReadCCConfigApi	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable the optional Frlf_ReadCCConfig API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06117]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfReconfigLPduSupport	EcucBooleanParamDef
BSW Description	
Configuration parameter to enable/disable Frlf support to enable/disable the reconfiguration of a given LPdu according to the parameters (FrameId, Channel, CycleRepetition, CycleOffset, PayloadLength, HeaderCRC) at runtime.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06109]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfTxConflictNotificationHeaderName	EcucStringParamDef
BSW Description	
Configuration of the header file name that defines the UL_TxConflictNotification.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06123]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfTxConflictNotificationName	EcucStringParamDef
BSW Description	
Configuration of the API name that is called in case a TxConflict has been detected.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06122]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfUnusedBitValue	EcucIntegerParamDef
BSW Description	
Set unused bits of transmitted Pdus to a defined value.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_00001]

BSW Module	BSW Context
Frlf	Frlf/FrlfGeneral
BSW Parameter	BSW Type
FrlfVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables/disables the existence of the Frlf_GetVersionInfo() API service	
true: Frlf_GetVersionInfo() API service exists false: Frlf_GetVersionInfo() API service does not exist	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_FrIf_06083]
-------	-------------------

C.4.3 FrNm Mapping

BSW Module	BSW Context	
FrNm	FrNm	
BSW Parameter	BSW Type	
FrNmChannelConfig	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters for all FlexRay NM channels.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00002]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig	
BSW Parameter	BSW Type	
FrNmChannel	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration parameters for a FlexRay NM Channel.		
Template Description		
FlexRay specific NM cluster attributes.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster		
Mapping Rule		Mapping Type
Create Container for each existing FlexrayNmCluster.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00006]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel	
BSW Parameter	BSW Type	
FrNmChannelIdentifiers	EcucParamConfContainerDef	
BSW Description		
This container contains instance specific identifiers related to the respective FlexRay Channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00007]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter	BSW Type	
FrNmActiveWakeupBitEnabled	EcucBooleanParamDef	
BSW Description	Enables/Disables the handling of the Active Wakeup Bit in the FrNm module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00082]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter	BSW Type	
FrNmCarWakeUpBitPosition	EcucIntegerParamDef	
BSW Description	Specifies the Bit position of the CWU within the NM-Message.	
Template Description	Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	Mapping Type	
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCarWakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00076]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter	BSW Type	
FrNmCarWakeUpBytePosition	EcucIntegerParamDef	
BSW Description	Specifies the Byte position of the CWU within the NM-Message.	
Template Description	Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	Mapping Type	
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCarWakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00075]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmCarWakeUpFilterEnabled		EcucBooleanParamDef
BSW Description		
<p>If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilterNodeId is considered as CWU request.</p> <p>FALSE - CWU Filtering is not supported</p> <p>TRUE - CWU Filtering is supported</p>		
Template Description		
<p>If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.</p>		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpFilterEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00077]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmCarWakeUpFilterNodeId		EcucIntegerParamDef
BSW Description		
<p>Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilterNodeId is considered as CWU request.</p>		
Template Description		
<p>Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.</p>		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpFilterNodeId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00078]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmCarWakeUpRxEnabled		EcucBooleanParamDef
BSW Description		
<p>Enables or disables support of CarWakeUp bit evaluation in received NM messages.</p> <p>FALSE - CarWakeUp not supported</p> <p>TRUE - CarWakeUp supported</p>		
Template Description		
<p>If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.</p>		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpRxEnabled		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00074]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmChannelHandle		EcucReferenceDef
BSW Description		
Channel identifier configured for the respective instance of the NM.		
The FrNmChannelHandle shall be encoded in the FrNmRxPduld parameter which is passed to FrNm_RxIndication() function called by the FrIf.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00013]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmComMNetworkHandleRef		EcucReferenceDef
BSW Description		
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00014]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
This parameter is used to enable or disable node detection support for a FrNm Channel.		
Template Description		
Enables the Request Repeat Message Request support. Only valid if nmNodeEnabled is set to true.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNodeDetectionEnabled		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00086]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmNodeId		EcucIntegerParamDef
BSW Description		
<p>NM node identifier configured for the respective FlexRay Channel.</p> <p>It is used for identifying the respective NM node in the NM-cluster. It must be unique for each NM node within one NM cluster.</p>		
Template Description		
Node identifier of local NmNode. Shall be unique in the NmCluster.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmNodeId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00017]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmPduScheduleVariant		EcucEnumerationParamDef
BSW Description		
<p>This parameter defines the PDU scheduling variant that should be used for this channel.</p> <p>Option 1 NM-Vote and NM-Data in static segment (one PDU) Option 2 NM-Vote and NM-Data in dynamic segment (one PDU) Option 3 NM-Vote and NM-Data in static segment (separate PDU) Option 4 NM-Vote in static segment and NM-Data in dynamic segment Option 5 NM-Vote in dynamic segment and NM-Data in static segment Option 6 NM-Vote and NM-Data in dynamic segment (separate PDU) Option 7 Combined NM-Vote and CBV in static segment and NM-Data in dynamic segment</p>		
Template Description		
FrNm schedule variant according to FrNm SWS.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmClusterCoupling.nmScheduleVariant		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00022]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmPnEnabled		EcucBooleanParamDef
BSW Description		

Enables or disables support of partial networking.	
false: Partial networking Range not supported true: Partial networking supported	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmPncParticipation	
Mapping Rule	Mapping Type
If NmCluster.nmPncParticipation has the value "true" or is not defined then FrNmPnEnabled shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00072]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	BSW Type
FrNmPnEraCalcEnabled	EcucBooleanParamDef
BSW Description	
Specifies if FrNm calculates the PN request information for external requests. (ERA)	
false: PN request are not calculated true: PN request are calculated	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00071]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	BSW Type
FrNmPnEraRxNSduRef	EcucReferenceDef
BSW Description	
Reference to a global Pdu. The SduRef is required for every FrNm Channel, because ERA is reported per channel.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00070]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers

BSW Parameter		BSW Type	
FrNmRxPdu		EcucParamConfContainerDef	
BSW Description			
This container describes the FlexRay NM RX PDU:s.			
Template Description			
M2 Parameter			
SystemTemplate::NetworkManagement::NmNode.rxNmPdu			
Mapping Rule			Mapping Type
Create Container if the regarded NmNode recieves a Pdu			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00010]	

BSW Module		BSW Context	
FrNm		FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		BSW Type	
FrNmRxPduContainsData		EcucBooleanParamDef	
BSW Description			
This parameter defines if the PDU contains NM Data.			
Template Description			
Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmDataInformation, System Template::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping			
Mapping Rule			Mapping Type
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00027]	

BSW Module		BSW Context	
FrNm		FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		BSW Type	
FrNmRxPduContainsVote		EcucBooleanParamDef	
BSW Description			
This parameter defines if the PDU contains NM Vote information.			
Template Description			
Defines if the Pdu contains NM Vote information.			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmVoteInformation			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00026]	

BSW Module	BSW Context
-------------------	--------------------

FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		BSW Type
FrNmRxPduld	EcucIntegerParamDef	
BSW Description		
PDU identifier configured for the respective FlexRay Channel.		
It is used for referring to the FlexRay Interface receive function. It must be consistent with the value configured in the FlexRay Interface. This ID is used for the combined reception of NM Vote and NM Data or for the reception of the NM Vote if NM Data is received in a separate PDU.		
ImplementationType: PduldType		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00025]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		BSW Type
FrNmRxPduRef	EcucReferenceDef	
BSW Description		
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the FrLf module to derive the PDU Id.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00012]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmSourceNodeIdentifierEnabled	EcucBooleanParamDef	
BSW Description		
This parameter is used to enable or disable SourceNodeIdentifier support for a FrNm Channel.		
Template Description		
Enables the source node identifier.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNodeIdEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00085]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter	BSW Type	
FrNmSynchronizationPointEnabled	EcucBooleanParamDef	
BSW Description		
This parameter defines if this channel shall provide the synchronization point indication to the NM Interface.		
Template Description		
If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmSynchronizingNetwork		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00021]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter	BSW Type	
FrNmTxPdu	EcucParamConfContainerDef	
BSW Description		
This container describes the FlexRay NM TX PDU:s.		
Template Description		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule	Mapping Type	
Create Container if the regarded NmNode transmits a Pdu	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00009]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter	BSW Type	
FrNmTxConfirmationPduId	EcucIntegerParamDef	
BSW Description		
Handle Id used by the Lower Layer when calling FrNm_TriggerTransmit() or FrNm_TxConfirmation().		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00018]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter	BSW Type	

FrNmTxPduContainsData	EcucBooleanParamDef
BSW Description	
This parameted defines if the PDU contains NM Data.	
Template Description	
Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmDataInformation, System Template::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping	
Mapping Rule	Mapping Type
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00024]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu
BSW Parameter	BSW Type
FrNmTxPduContainsVote	EcucBooleanParamDef
BSW Description	
This parameted defines if the PDU contains NM Vote information.	
Template Description	
Defines if the Pdu contains NM Vote information.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmVoteInformation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00023]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu
BSW Parameter	BSW Type
FrNmTxPduRef	EcucReferenceDef
BSW Description	
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference is used to derive the PDU Id that is defined by the FrIf module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00011]

BSW Module	BSW Context
-------------------	--------------------

FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		BSW Type
FrNmUserDataTxPdu	EcucParamConfContainerDef	
BSW Description		
This optional container is used to configure the UserNm PDU. This container is only available if FrNmComUserDataSupport is enabled.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		Mapping Type
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00055]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmUserDataTxPdu	
BSW Parameter		BSW Type
FrNmTxUserDataPduId	EcucIntegerParamDef	
BSW Description		
This parameter defines the Handle ID of the NM User Data I-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00056]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmUserDataTxPdu	
BSW Parameter		BSW Type
FrNmTxUserDataPduRef	EcucReferenceDef	
BSW Description		
Reference to the NM User Data I-PDU in the global PDU collection.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00057]

BSW Module	BSW Context
-------------------	--------------------

FrNm	FrNm/FrNmChannelConfig/FrNmChannel	
BSW Parameter		BSW Type
FrNmChannelTiming	EcucParamConfContainerDef	
BSW Description		
This container contains instance-specific timing related to the respective FlexRay Channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00008]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmDataCycle	EcucEnumerationParamDef	
BSW Description		
Number of FlexRay Schedule Cycles needed to transmit the NM Data of all ECUs on the FlexRay bus		
Template Description		
Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmDataCycle		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00031]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
This parameter defines the processing cycle of the main function of FrNm module in seconds.		
Template Description		
Defines the processing cycle of the main function of FrNm module.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmMainFunctionPeriod		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00035]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmReadySleepCnt	EcucIntegerParamDef	

BSW Description	
FrNm switches to bus sleep mode at the end of the FrNmReadySleepCnt+1 repetition cycle without any NM vote. E.g. on a value of "1", the NM-State Machine will leave the Ready Sleep State after two NM Repetition Cycles with no "keep awake" votes.	
Template Description	
The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.nmReadySleepTime	
Mapping Rule	Mapping Type
FrNmReadySleepCnt = ((Float2Int(nmReadySleepTime/cycle))/nmRepetitionCycle)-1	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00051]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming
BSW Parameter	BSW Type
FrNmRemoteSleepIndTime	EcucFloatParamDef
BSW Description	
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep.	
The value "0" denotes that no Remote Sleep Indication functionality is configured.	
Template Description	
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.	
M2 Parameter	
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRemoteSleepIndicationTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00029]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming
BSW Parameter	BSW Type
FrNmRepeatMessageTime	EcucFloatParamDef
BSW Description	
Timeout for Repeat Message State. Defines the time in seconds how long the NM shall stay in the Repeat Message State.	
The value "0" denotes that no Repeat Message State is configured, which means that Repeat Message State is transient and implies that it is left immediately after entry and consequently no startup stability is guaranteed and no node detection procedure is possible.	
Template Description	
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
M2 Parameter	
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00030]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmRepetitionCycle		EcucEnumerationParamDef
BSW Description		
Number of Flexray Schedule Cycles used to repeat the transmission of the Nm vote of all ECUs on the Flexray Bus.		
Template Description		
Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all FlexRay NmEcus of this FlexRayNmCluster. This value shall be an integral multiple of nmVotingCycle.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRepetitionCycle		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00033]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmVoteInhibitionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the inhibition of vote changes from the next-to-last repetition cycle to the last repetition cycle before the Ready Sleep Counter expires.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00053]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		BSW Type
FrNmVotingCycle		EcucEnumerationParamDef
BSW Description		
Number of FlexRay Schedule Cycles needed to transmit the Nm vote of all ECUs on the FlexRay Bus.		
Template Description		
Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.		
M2 Parameter		

SystemTemplate::NetworkManagement::FlexrayNmCluster.nmVotingCycle	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00032]

BSW Module	BSW Context
FrNm	FrNm
BSW Parameter	BSW Type
FrNmGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains all global configuration parameters for the FrNm module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00001]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig
BSW Parameter	BSW Type
FrNmGlobalFeatures	EcucParamConfContainerDef
BSW Description	
This container contains module features related to the FlexRay NM functionality.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00004]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmBusSynchronizationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the bus synchronization.	
Template Description	
Enables bus synchronization support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00048]

BSW Module		BSW Context	
FrNm		FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type	
FrNmComUserDataSupport		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.			
Template Description			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping			
Mapping Rule			Mapping Type
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00054]	

BSW Module		BSW Context	
FrNm		FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type	
FrNmCoordinatorSyncSupport		EcucBooleanParamDef	
BSW Description			
Enables/disables the coordinator synchronization support.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00081]	

BSW Module		BSW Context	
FrNm		FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type	
FrNmCycleCounterEmulation		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling the cycle counter emulation.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrNm_00060]	

BSW Module		BSW Context	
FrNm		FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	

BSW Parameter		BSW Type
FrNmDualChannelPduEnable		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the support of dual channel transmission and reception of NM messages.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00049]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type
FrNmHwVoteEnable		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the processing of FlexRay Hardware aggregated NM-Votes. This switch enables/disables the optional API FrIf_GetNmVector.		
Template Description		
Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.		
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmEcu.nmHwVoteEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00050]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type
FrNmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling Passive Node Configuration support.		
Template Description		
Enables support of the Passive Mode. The passive mode is configurable per channel.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmPassiveModeEnabled		
Mapping Rule		Mapping Type
1:1 mapping. nmNode.nmPassiveModeEnabled shall always have the same value in all Nm Clusters with the same bus protocol in the scope of one EcuInstance.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00043]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		BSW Type

FrNmPduRxIndicationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling PDU reception indication.	
Template Description	
Switch for enabling the PDU Rx Indication.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00046]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmPnEiraCalcEnabled	EcucBooleanParamDef
BSW Description	
Specifies if FrNm calculates the PN request information for internal an external requests. (EIRA) true: PN request are calculated false: PN request are not calculated	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00067]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmPnEiraRxNSduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack. Only one SduRef is required for FrNm because the EIRA is the aggregation over all FlexRay Channels.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00069]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmPnInfo	EcucParamConfContainerDef

BSW Description	
PN information configuration	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00061]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter		BSW Type
FrNmPnFilterMaskByte		EcucParamConfContainerDef
BSW Description		
Filter mask byte configuration		
Template Description		
Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		Mapping Type
<p>For one EcuInstance all contributing FlexrayCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.</p>		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00064]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo/FrNmPnFilterMaskByte	
BSW Parameter		BSW Type
FrNmPnFilterMaskByteIndex		EcucIntegerParamDef
BSW Description		
Index of the filter mask byte. Specifies the position within the filter mask byte array.		
Template Description		
Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		Mapping Type

For one EcuInstance all contributing FlexrayCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00065]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo/FrNmPnFilterMaskByte	
BSW Parameter	BSW Type	
FrNmPnFilterMaskByteValue	EcuIntegerParamDef	
BSW Description		
Parameter to configure the filter mask byte.		
Template Description		
Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		Mapping Type
For one EcuInstance all contributing FlexrayCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00066]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter	BSW Type	
FrNmPnInfoLength	EcuIntegerParamDef	
BSW Description		
Specifies the length of the PN request information in the NM message.		
Template Description		
Length of the partial networking request release information vector (in bytes).		
M2 Parameter		
SystemTemplate::System.pncVectorLength		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00063]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter	BSW Type	
FrNmPnInfoOffset	EcucIntegerParamDef	
BSW Description		
Specifies the offset of the PN request information in the NM message.		
Template Description		
Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.		
M2 Parameter		
SystemTemplate::System.pncVectorOffset		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00062]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter	BSW Type	
FrNmPnResetTime	EcucFloatParamDef	
BSW Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA. The value shall be the same for every channel. Thus it is a global config parameter.		
Template Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcucInstance.pnResetTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00068]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter	BSW Type	
FrNmRemoteSleepIndicationEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling remote sleep indication.		
calculationFormula = If (FrNmPassiveModeEnabled == True) then Equal(False) else Equal(False or True)		
Template Description		
Switch for enabling remote sleep indication support.		
M2 Parameter		

SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00044]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmStateChangeIndicationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling state change indication.	
Template Description	
Enables the CAN Network Management state change notification.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00047]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmUserDataEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling user data support.	
Template Description	
Switch for enabling user data support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00039]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	BSW Type
FrNmVotingNextToLastRepetitionCycleDisable	EcucBooleanParamDef
BSW Description	
Pre-processor switch for disabling vote changes in the last two repetition cycles before the Ready Sleep Counter expires.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_FrNm_00073]
-------	-------------------

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig	
BSW Parameter	BSW Type	
FrNmGlobalProperties	EcucParamConfContainerDef	
BSW Description		
This container contains module properties related to the FlexRay NM functionality.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00003]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter	BSW Type	
FrNmDevErrorDetect	EcucBooleanParamDef	
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrNm_00036]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter	BSW Type	
FrNmMainAcrossFrCycle	EcucBooleanParamDef	
BSW Description		
If the FlexRay NM MainFunction is executed completely within the FlexRay communication cycle where the last NM vote of the current vote cycle is received, the FrNmMainAcrossFrCycle shall be configured to FALSE.		
If the FlexRay NM MainFunction is executed completely within the FlexRay communication cycle subsequent to the one where the last NM vote of the current vote cycle is received, the FrNmMainAcrossFrCycle shall be configured to TRUE.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00038]	

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter		BSW Type
FrNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrNm_00037]	

C.4.4 FrTp Mapping

BSW Module	BSW Context	
FrTp	FrTp	
BSW Parameter		BSW Type
FrTpGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration parameters of the FlexRay Transport Protocol module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00009]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		BSW Type
FrTpAckRt		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling the Acknowledgement and retry mechanisms.		
True: Acknowledge and Retry is enabled False: Acknowledge and Retry is disabled		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00002]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		BSW Type
FrTpChanNum		EcucIntegerParamDef
BSW Description		
Preprocessor switch for defining the number of concurrent channels the module supports. Up to 32 channels shall be definable here.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00004]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		BSW Type
FrTpChangeParamApi		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling the API to change FrTp communication parameters. True: ChangeParameter API is enabled False: ChangeParameter API is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00052]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		BSW Type
FrTpDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00008]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	BSW Type
FrTpFullDuplexEnable	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling full duplex mechanisms for all channels. True: Full duplex is enabled False: Full duplex is disabled (Half duplex is enabled)	
Template Description	
The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpEcu.fullDuplexEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00051]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	BSW Type
FrTpMainFuncCycle	EcucFloatParamDef
BSW Description	
This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.	
Template Description	
The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpEcu.cycleTimeMainFunction	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00011]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	BSW Type
FrTpTransmitCancellation	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation. True: Transmit/Receive Cancellation is enabled False: Transmit/Receive Cancellation is disabled	
Template Description	

With this switch Tx and Rx Cancellation can be turned on or off.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpEcu.cancellation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00036]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	BSW Type
FrTpUnknownMsgLength	EcucBooleanParamDef
BSW Description	
Preprocessor switch to support data transfer with unknown message length.	
True: Transmission with unknown message length is enabled False: Transmission with unknown message length is disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00044]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	BSW Type
FrTpVersionInfoApi	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling the Version info API.	
True: Version Info API is enabled False: Version Info API is disabled	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00045]

BSW Module	BSW Context
FrTp	FrTp
BSW Parameter	BSW Type
FrTpMultipleConfig	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR FrTp module.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00018]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig
BSW Parameter	BSW Type
FrTpConnection	EcucParamConfContainerDef
BSW Description	
This container contains the connection specific parameters to transfer N-PDUs via FlexRay TP.	
Template Description	
A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection.	
In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection	
Mapping Rule	Mapping Type
Create container for each FlexRayTpConnection that is described in the ECU Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00006]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type
FrTpBandwidthLimitation	EcucBooleanParamDef
BSW Description	
This parameter indicates whether the connection requires a bandwidth limitation or not. If FrTpBandwidthLimitation=True the sender shall send a StartFrame always on the first PDU of a PDU-Pool.	
Template Description	
Specifies whether the connection requires a bandwidth limitation or not.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.bandwidthLimitation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00050]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type

FrTpConCtrlRef	EcucReferenceDef
BSW Description	
FrTpConnectionControlReference: This parameter defines a reference to a connection control container.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.tpConnectionControl	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00005]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type
FrTpLa	EcucIntegerParamDef
BSW Description	
This parameter defines the Local Address for the respective connection. When the local instance is the sender, this is the Source Address within the TP frame. When the local instance is the receiver, this is the Target Address within the TP frame. If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.transmitter, SystemTemplate::TransportProtocols::FlexrayTpConnection.receiver	
Mapping Rule	Mapping Type
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00010]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type
FrTpMultipleReceiverCon	EcucBooleanParamDef
BSW Description	
This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection. If data segmentation is required this parameter is used to check whether segmentation is possible or not. If the connection is 1:n segmentation is not possible and an error will occur.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.multicast	
Mapping Rule	Mapping Type
If FlexRayTpConnection contains a mulicast reference to TpAddress than set this parameter to true	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00019]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter	BSW Type	
FrTpRa	EcucIntegerParamDef	
BSW Description		
This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame. If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.transmitter, SystemTemplate::Transport Protocols::FlexrayTpConnection.receiver		
Mapping Rule		Mapping Type
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00021]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter	BSW Type	
FrTpRxpduPoolRef	EcucReferenceDef	
BSW Description		
This parameter defines a reference to a RxPduPool.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.rxPduPool, SystemTemplate::Transport Protocols::FlexrayTpConnection.txPduPool		
Mapping Rule		Mapping Type
Depending whether the regarded Ecu is the transmitter or the receiver this reference shall be created if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the txPduPool or rxPduPool reference.		full
If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus.		
If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.		
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00025]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter	BSW Type	
FrTpRxsdu	EcucParamConfContainerDef	
BSW Description		

This parameter defines the Rx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR. This N-SDU can produce meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.

Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.directTpSdu	
Mapping Rule	Mapping Type
Create container if an Rx Pdu is referenced by the FlexRayTpConnection	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00027]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpRxSdu
BSW Parameter	BSW Type
FrTpRxSduId	EcucIntegerParamDef
BSW Description	
This unique identifier is used for change parameter request or receive cancellation from PduR to FrTp.	
ImplementationType: PduIdType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00053]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpRxSdu
BSW Parameter	BSW Type
FrTpRxSduRef	EcucReferenceDef
BSW Description	
Reference to a PDU in the global PDU structure.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00028]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type
FrTpTxPduPoolRef	EcucReferenceDef
BSW Description	

This parameter defines a reference to a TxPduPool.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.rxPduPool, SystemTemplate::TransportProtocols::FlexrayTpConnection.txPduPool	
Mapping Rule	Mapping Type
Depending whether the regarded Ecu is the transmitter or the receiver this reference shall be created if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the txPduPool or rxPduPool reference. If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus. If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00039]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	BSW Type
FrTpTxSdu	EcucParamConfContainerDef
BSW Description	
This parameter defines the Tx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR. This N-SDU can consume meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnection.directTpSdu	
Mapping Rule	Mapping Type
Create container if an Tx Pdu is referenced by the FlexRayTpConnection	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00041]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpTxSdu
BSW Parameter	BSW Type
FrTpTxSdulD	EcucIntegerParamDef
BSW Description	
This is a unique identifier for a to be transmitted message from the PduR to the FrTp. ImplementationType: PdulDType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00042]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpTxSdu	
BSW Parameter		BSW Type
FrTpTxSduRef		EcucReferenceDef
BSW Description		
Reference to a PDU in the global PDU structure.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00043]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		BSW Type
FrTpConnectionControl		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters to control a FlexRay TP connection.		
Template Description		
Configuration parameters to control a FlexRay TP connection.		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl		
Mapping Rule		Mapping Type
Create container for each FlexRayTpConnectionControl that is described in the ECU Extract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00007]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		BSW Type
FrTpAckType		EcucEnumerationParamDef
BSW Description		
This parameter defines the type of acknowledgement which is used for the specific channel.		
Template Description		
This parameter defines the type of acknowledgement which is used for the specific channel.		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.ackType		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00003]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		BSW Type
FrTpMaxFCWait		EcucIntegerParamDef

BSW Description	
This parameter defines the maximum number of FlowControl N-PDUs with FlowState "WAIT"	
Template Description	
This attribute defines the maximum number of FlowControl N-PDUs with FlowState "WAIT".	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxFcWait	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00014]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpMaxNbrOfNPduPerCycle	EcucIntegerParamDef
BSW Description	
This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
Template Description	
This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxNumberOfNpduPerCycle	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00029]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpMaxRn	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of retries (if retry is configured).	
Template Description	
This parameter defines the maximum number of retries (if retry is configured for the particular channel).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxRetries	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00017]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpSCexp	EcucIntegerParamDef
BSW Description	

This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It represents the exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.	
Template Description	
Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.separationCycleExponent	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00020]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpTimeBr	EcucFloatParamDef
BSW Description	
This parameter defines the time in seconds the FrTp requires to transmit a corresponding FlowControl Frame. According to ISO 10681-2 this parameter is a performance requirement.	
Template Description	
Time (in seconds) until transmission of the next FlowControl N-PDU.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeBr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00047]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpTimeCs	EcucFloatParamDef
BSW Description	
This parameter defines the time in seconds between the sending of two CFs or between the sending of a CF and LF or between the reception of a FC and sending of the next CF.	
Template Description	
Time (in seconds) until transmission of the next ConsecutiveFrame NPdu / LastFrame NPdu.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeCs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00056]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpTimeoutAr	EcucFloatParamDef
BSW Description	

This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
Template Description	
This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutAr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00032]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpTimeoutAs	EcucFloatParamDef
BSW Description	
This parameter specifies the timeout in seconds the FrIf shall confirm a transmitted Pdu to the FrTp.	
Template Description	
This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutAs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00033]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	BSW Type
FrTpTimeoutBs	EcucFloatParamDef
BSW Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
Template Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutBs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00034]

BSW Module	BSW Context
-------------------	--------------------

FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		BSW Type
FrTpTimeoutCr	EcucFloatParamDef	
BSW Description		
This parameter defines the timeout value in seconds a receiver is waiting for a CF or a LF.		
Template Description		
This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutCr		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00035]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		BSW Type
FrTpMaxConnectionCnt	EcucIntegerParamDef	
BSW Description		
Maximum number of TP connections. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxConnectionCnt		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00054]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		BSW Type
FrTpRxPduPool	EcucParamConfContainerDef	
BSW Description		
This container contains all Pdus that are assigned to that Pdu Pool.		
Template Description		
FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpPduPool		
Mapping Rule		Mapping Type
Create container if the FlexrayTpPduPool element is referenced by the Flexray TpConnection via the rxPduPool or txPduPool reference.		full
If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus.		
If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.		
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTp_00024]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool	
BSW Parameter	BSW Type	
FrTpRxPdu	EcucParamConfContainerDef	
BSW Description	Container to hold the PDU parameters.	
ImplementationType: PduInfoType		
Template Description		
M2 Parameter	SystemTemplate::TransportProtocols::FlexrayTpPduPool.nPdu	
Mapping Rule	Mapping Type	
Create container for each NPdu that is referenced by the regarded FlexrayTpPduPool.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00022]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool/FrTpRxPdu	
BSW Parameter	BSW Type	
FrTpRxPduld	EcucIntegerParamDef	
BSW Description	This is a unique identifier for a received message which is forwarded from the FrIf to the FrTp.	
ImplementationType: PduldType		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00023]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool/FrTpRxPdu	
BSW Parameter	BSW Type	
FrTpRxPduRef	EcucReferenceDef	
BSW Description	Reference to a PDU in the global PDU structure.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00026]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter	BSW Type	
FrTpTxPduPool	EcucParamConfContainerDef	
BSW Description	This container contains all Pdus that are assigned to that Pdu Pool.	
Template Description	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.	
M2 Parameter	SystemTemplate::TransportProtocols::FlexrayTpPduPool	
Mapping Rule	Mapping Type	
Create container if the FlexrayTpPduPool element is referenced by the Flexray TpConnection via the rxPduPool or txPduPool reference.	full	
If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus.		
If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00038]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool	
BSW Parameter	BSW Type	
FrTpTxPdu	EcucParamConfContainerDef	
BSW Description	Container to hold the PDU parameters.	
ImplementationType: PduInfoType		
Template Description		
M2 Parameter	SystemTemplate::TransportProtocols::FlexrayTpPduPool.nPdu	
Mapping Rule	Mapping Type	
Create container for each NPdu that is referenced by the regarded FlexrayTp PduPool.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00037]	

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool/FrTpTxPdu	
BSW Parameter	BSW Type	
FrTpTxConfirmationPduld	EcucIntegerParamDef	
BSW Description	Handle Id to be used by the Frlf to confirm the transmission of the FrTpTxPdu to the Frlf module (FrTp_TxConfirmation) and for TriggerTransmit (FrTp_TriggerTransmit).	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTp_00049]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool/FrTpTxPdu	
BSW Parameter	BSW Type	
FrTpTxPduRef	EcucReferenceDef	
BSW Description	Reference to a PDU in the global PDU structure.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTp_00040]	

C.4.5 FrArTp Mapping

BSW Module	BSW Context	
FrArTp	FrArTp	
BSW Parameter	BSW Type	
FrArTpGeneral	EcucParamConfContainerDef	
BSW Description	This container contains the general configuration (parameters) of the FlexRay TP.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00012]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter	BSW Type	
FrArTpDevErrorDetect	EcucBooleanParamDef	
BSW Description	Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_FrArTp_00011]
-------	---------------------

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter	BSW Type	
FrArTpHaveAckRt	EcucBooleanParamDef	
BSW Description	Preprocessor switch for enabling the Acknowledgement and retry mechanisms.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00014]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter	BSW Type	
FrArTpHaveGrpSeg	EcucBooleanParamDef	
BSW Description	Preprocessor switch for enabling segmentation of 1:n messages.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00015]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter	BSW Type	
FrArTpHaveLm	EcucBooleanParamDef	
BSW Description	Preprocessor switch for enabling the mechanism for message longer than allowed by.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00016]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	

BSW Parameter		BSW Type	
FrArTpHaveTc		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrArTp_00017]	

BSW Module		BSW Context	
FrArTp		FrArTp/FrArTpGeneral	
BSW Parameter		BSW Type	
FrArTpMainFuncCycle		EcucFloatParamDef	
BSW Description			
This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrArTp_00020]	

BSW Module		BSW Context	
FrArTp		FrArTp/FrArTpGeneral	
BSW Parameter		BSW Type	
FrArTpVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling the Version info API.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrArTp_00054]	

BSW Module		BSW Context	
FrArTp		FrArTp	
BSW Parameter		BSW Type	
FrArTpMultipleConfig		EcucParamConfContainerDef	
BSW Description			

This container contains the configuration parameters and sub containers of the AUTOSAR FrArTp module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00028]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig
BSW Parameter	BSW Type
FrArTpChannel	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of one FlexRay TP channel.	
Template Description	
A channel is a group of connections sharing several properties.	
The FlexRay AutosarTransport Layer supports several channels. These channels can work concurrently, thus each of them requires its own state machine and management data structures and its own PDU-IDs.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel	
Mapping Rule	Mapping Type
Create container for each FlexrayArTpChannel that exists in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00005]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpAckType	EcucEnumerationParamDef
BSW Description	
This parameter defines the type of acknowledgement which is used for the specific channel.	
Template Description	
Type of Acknowledgement.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.ackType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00002]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpAdrType	EcucEnumerationParamDef
BSW Description	

This parameter states the addressing type this connection has. The meanings of the values are one byte and two byte.	
Template Description	
Addressing Type of this connection:	
true: Two Bytes	
false: One Byte	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.extendedAddressing	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00008]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpConcurrentConnections	EcucIntegerParamDef
BSW Description	
This parameter defines the number of connections that can be active at the same time. If set to 0, all configured connections can be active at the same time.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00057]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpConnection	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of one FlexRay TP connection.	
A connection can only belong to one channel.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.tpConnection	
Mapping Rule	Mapping Type
Create container for each existing FlexrayArTpConnection that is aggregated by FlexrayArTpChannel in the System description.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00010]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection

BSW Parameter		BSW Type	
FrArTpConPrioPdu		EcucIntegerParamDef	
BSW Description			
This parameter defines the number of TxNPdus to which this connection has prioritized access. It must be ensured that the number of prioritized PDUs of all connections is smaller than the total number of TxNPdus in the associated PDU pool.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrArTp_00058]	

BSW Module		BSW Context	
FrArTp		FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter		BSW Type	
FrArTpLa		EcucIntegerParamDef	
BSW Description			
<p>This parameter defines the Local Address for the respective connection.</p> <p>When the local instance is the sender, this is the Source Address within the TP frame.</p> <p>When the local instance is the receiver, this is the Target Address within the TP frame.</p> <p>Note that in case of 1 byte addressing only the values from 0x0000 - 0x00FF are valid.</p> <p>If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16.</p>			
Template Description			
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.source			
Mapping Rule			Mapping Type
LocalAddress can be derived from the TpNode that is referenced by the FlexRayTpConnection as source.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrArTp_00018]	

BSW Module		BSW Context	
FrArTp		FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter		BSW Type	
FrArTpMultRec		EcucBooleanParamDef	
BSW Description			
<p>This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection.</p> <p>Of course, if the channel to which the connection is configured has retry or acknowledgement enabled, no retry or acknowledgement will occur in case the connection is an 1:n connection.</p>			
Template Description			
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.multicast			
Mapping Rule			Mapping Type
If multicast is used set this attribute to true.			full

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00027]

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter	BSW Type	
FrArTpRa	EcucIntegerParamDef	
BSW Description		
<p>This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame. Note that in case of 1 byte addressing only the values from 0x0000 - 0x00FF are valid.</p> <p>If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16.</p>		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpConnection.target		
Mapping Rule		Mapping Type
RemoteAddress can be derived from the TpNode that is referenced by the FlexRayTpConnection as target.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00037]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter	BSW Type	
FrArTpRxSdu	EcucParamConfContainerDef	
BSW Description		
Describes the Rx N-SDU. This N-SDU can produce meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpConnection.directTpSdu		
Mapping Rule		Mapping Type
Create container for every IPdu that is received by the FrArTp and the regarded Ecu.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrArTp_00038]	

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpRxSdu	
BSW Parameter	BSW Type	
FrArTpRxSduRef	EcucReferenceDef	
BSW Description		
Reference to a PDU in the global PDU structure.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00039]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpRxSdu
BSW Parameter	BSW Type
FrArTpSduRxId	EcucIntegerParamDef
BSW Description	
This is a unique identifier for a received message. This Id is used in the CancelReceive and ChangeParameter API call.	
ImplementationType: PduldType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00040]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection
BSW Parameter	BSW Type
FrArTpTxSdu	EcucParamConfContainerDef
BSW Description	
Describes the Tx N-SDU. This N-SDU can consume meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpConnection.directTpSdu	
Mapping Rule	Mapping Type
Create container for every IPdu that is transmitted by the FrArTp and the regarded Ecu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00055]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpTxSdu
BSW Parameter	BSW Type
FrArTpSduTxId	EcucIntegerParamDef
BSW Description	

This is a unique identifier for a received or a to be transmitted message. With this (and by means of e.g. a lookup table) the PDU Router can route the message appropriately without dealing with the particularities of the Transport Layer. This parameter can also be seen as the identifier of a connection.	
ImplementationType: PduldType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00041]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpTxSdu
BSW Parameter	BSW Type
FrArTpTxSduRef	EcucReferenceDef
BSW Description	
Reference to a PDU in the global PDU structure.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00052]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpGrpSeg	EcucBooleanParamDef
BSW Description	
Here can be specified, whether segmentation within a 1:n connection is allowed or not.	
Template Description	
This attribute defines whether segmentation within a 1:n connection is allowed or not.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.multicastSegmentation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00013]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpLm	EcucEnumerationParamDef
BSW Description	

This specifies the maximum message length for the particular channel.	
Template Description	
This specifies the maximum message length for the particular channel.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maximumMessageLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00019]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpMaxAr	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs.	
Template Description	
This attribute defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxAr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00021]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpMaxAs	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs.	
Template Description	
This attribute defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxAs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00022]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpMaxBs	EcucIntegerParamDef
BSW Description	
This parameter defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.	
Template Description	

This attribute defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxBs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00023]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpMaxRn	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of retries (if retry is configured for the particular channel).	
Template Description	
This attribute defines the maximum number of retries (if retry is configured for the particular channel).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxRetries	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00026]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpMaxWft	EcucIntegerParamDef
BSW Description	
This parameter defines the maximal number of wait frames to be sent for a pending connection.	
Template Description	
This attribute defines the maximal number of wait frames to be sent for a pending connection. Range is 0..255.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxFcWait	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00059]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpPdu	EcucParamConfContainerDef
BSW Description	
Container to hold the PDU parameters.	
ImplementationType: PduInfoType	
Template Description	

M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.nPdu	
Mapping Rule	Mapping Type
Create container if NPdus are referenced by the FlexrayArTpChannel.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00029]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu
BSW Parameter	BSW Type
FrArTpPduDirection	EcucEnumerationParamDef
BSW Description	
This parameter defines the direction of the PDU.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort	
Mapping Rule	Mapping Type
The direction of the Npdu can be derived from the triggering elements that contain references to IN- and OUT-Ports.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00030]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu
BSW Parameter	BSW Type
FrArTpPduld	EcucIntegerParamDef
BSW Description	
This is the identifier of the FlexRay Interface PDUs (Fr N-PDU, Fr L-SDU) in which the Transport Layer Frames of this channel should be transmitted. For FrArTpPduDirection == FRARTP_RX, this parameter specifies the ID that is used by FrIf when calling FrArTp_RxIndication, while for FrArTpPduDirection == FRARTP_TX this ID is used by FrIf when calling FrArTp_TxConfirmation or FrArTp_TriggerTransmit.	
ImplementationType: PduldType	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00035]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu
BSW Parameter	BSW Type
FrArTpPduRef	EcucReferenceDef
BSW Description	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00036]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpStMin	EcucFloatParamDef
BSW Description	
<p>This parameter defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>FrArTpStMin must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $FrArTpStMin = n * FrIfGdCycle * m$, where n is an integer ≥ 0 and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.</p> <p>Please note: Due to the scheduling strategies of FrArTp, FrArTpStMin can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p>	
Template Description	
<p>This attribute defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>The minimumSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $minimumSeparationTime = n * cycle * m$, where n is an integer ≥ 0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.</p> <p>Please note: Due to the scheduling strategies of FrTp, minimumSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p> <p>Range: 0 .. 0.127</p>	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.minimumSeparationTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00042]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpStMinGrpSeg	EcucFloatParamDef

BSW Description	
<p>This parameter defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>FrArTpStMinGrpSeg must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $FrArTpStMinGrpSeg = n * FrIfGdCycle * m$, where n is an integer ≥ 0 and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.</p> <p>Please note: Due to the scheduling strategies of FrArTp, FrArTpStMinGrpSeg can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p>	
Template Description	
<p>This attribute defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.</p> <p>minimumMulticastSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. $minimumMulticastSeparationTime = n * cycle * m$, where n is an integer ≥ 0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.</p> <p>Please note: Due to the scheduling strategies of FrTp, minimumMulticastSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.</p> <p>Range: 0 .. 0.127</p>	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.minimumMulticastSeperationTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00060]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpTc	EcucBooleanParamDef
BSW Description	
With this switch Transmit Cancellation and Receive Cancellation can be turned on or off for this channel.	
Template Description	
With this switch Tx and Rx Cancellation can be turned on or off.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.cancellation	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00043]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type

FrArTpTimeBr	EcucFloatParamDef
BSW Description	
<p>This parameter defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.</p> <p>It is obvious that $FRARTP_TIME_BR + (FRARTP_TIMEOUT_AR * FRARTP_MAX_AR) < FRARTP_TIMEOUT_BS$ must hold (because the transmission duration on the bus has also to be considered).</p> <p>This parameter is defined in ISO 15765-2. It is contained in the configuration as a performance requirement.</p>	
Template Description	
<p>This attribute defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.</p>	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeBr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00044]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpTimeCs	EcucFloatParamDef
BSW Description	
<p>This parameter defines the time in seconds between the sending of two consecutive CFs or between reception of an FC or AF and sending of the next CF .</p> <p>It is obvious that $FRARTP_TIME_CS + (FRARTP_TIMEOUT_AS * FRARTP_MAX_AS) < FRARTP_TIMEOUT_CR$ must hold (because the transmission duration on the bus has also to be considered).</p> <p>This parameter is defined in ISO 15765-2. It is contained in the configuration as a performance requirement.</p>	
Template Description	
<p>This attribute defines the time in seconds between the sending of two consecutive frames or between a consecutive frame and a flow control (for Transmit Cancellation) or between reception of a flow control or Acknowledgement Frame and sending of the next consecutive frame or a flow control (for Transmit Cancellation).</p>	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeCs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00046]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpTimeoutAr	EcucFloatParamDef
BSW Description	

This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
Template Description	
This attribute states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutAr	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00048]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpTimeoutAs	EcucFloatParamDef
BSW Description	
This parameter states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).	
Template Description	
This attribute states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutAs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00049]

BSW Module	BSW Context
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	BSW Type
FrArTpTimeoutBs	EcucFloatParamDef
BSW Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
Template Description	
This attribute defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutBs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrArTp_00050]

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		BSW Type
FrArTpTimeoutCr		EcucFloatParamDef
BSW Description		
This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.		
Template Description		
This attribute defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.		
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutCr		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrArTp_00051]

C.4.6 FrSM Mapping

BSW Module	BSW Context	
FrSM	FrSM	
BSW Parameter		BSW Type
FrSMConfig		EcucParamConfContainerDef
BSW Description		
This container comprises the cluster specific configuration of the FlexRay State Manager.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrSM_00146]

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig	
BSW Parameter		BSW Type
FrSMCluster		EcucParamConfContainerDef
BSW Description		
This container specifies a FlexRay cluster and all related data. A FlexRay cluster may consist of more than one controller per ECU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrSM_00067]

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMCheckWakeupReason		EcucBooleanParamDef	
BSW Description			
If FrSMCheckWakeupReason is true, the FrSM will check the wakeup reason in order to skip the wakeup in case of wakeup by bus. If FrSMCheckWakeupReason is false, the FrSM will always try to perform a wakeup.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00001]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMClusterDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00163]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster/FrSMClusterDemEventParameterRefs	
BSW Parameter		BSW Type	
FRSM_E_CLUSTER_STARTUP		EcucReferenceDef	
BSW Description			
Reference to the DemEventParameter which shall be issued when the error "FRSM_E_CLUSTER_STARTUP" has occurred. If the reference is not configured the error shall be reported as DET error.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00164]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster/FrSMClusterDemEventParameterRefs	
BSW Parameter		BSW Type	
FRSM_E_CLUSTER_SYNC_LOSS		EcucReferenceDef	
BSW Description			
Reference to the DemEventParameter which shall be issued when the error "FRSM_E_CLUSTER_SYNC_LOSS" has occurred. If the reference is not configured the error shall be reported as DET error.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00169]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMComMNetworkHandleRef		EcucReferenceDef	
BSW Description			
Reference to the unique handle to identify one certain FlexRay network correspond to one of the network handles of the ComM configuration.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00070]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMDelayStartupWithoutWakeup		EcucBooleanParamDef	
BSW Description			
If true, timer t1 shall be started instead of immediately calling FrIf_AllowColdstart in case of a startup without wakeup.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00166]	

BSW Module	BSW Context
------------	-------------

FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMDurationT1	EcucFloatParamDef	
BSW Description		
The duration of timer t1 in seconds. A value of 0 shall imply that the timer is not used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid	[ECUC_FrSM_00102]	

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMDurationT2	EcucFloatParamDef	
BSW Description		
The duration of timer t2 in seconds. A value of 0 shall imply that the timer is not used. The value of this parameter shall be larger than the value of FrSMDurationT1 parameter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid	[ECUC_FrSM_00089]	

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMDurationT3	EcucFloatParamDef	
BSW Description		
The duration of timer t3 in seconds. The value of this parameter shall be larger than the value of FrSMDurationT1 parameter. A value of 0 shall imply that the timer is not used. It shall only be possible to configure a value 0 if no FrNm is used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid	[ECUC_FrSM_00162]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMDurationT4		EcucFloatParamDef	
BSW Description			
The timer t4 ensures that a dual channel node will eventually clear its coldstart inhibit bit and become a leading coldstarter.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00173]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMFrlfClusterRef		EcucReferenceDef	
BSW Description			
References the cluster configuration in the FlexRay Interface configuration. Note that the assigned controllers and transceivers are defined in the Frlf configuration and can be accessed via this reference.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00116]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type	
FrSMIsColdstartEcu		EcucBooleanParamDef	
BSW Description			
True: The ECU is a coldstart node for this FlexRay cluster. False: The ECU is no coldstart node for this FlexRay cluster.			
Template Description			
FlexrayCommunicationController.keySlotID: ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedForStartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory.			
FlexrayCommunicationController.keySlotUsedForStartUp: Flag indicating whether the Key Slot is used to transmit a startup frame.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotID, SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotUsedForStartUp,			

Mapping Rule		Mapping Type
<= TRUE if keySlotId existing and valid (i.e. not 0) and keySlotUsedForStartUp set to true <= FALSE otherwise		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrSM_00068]	

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMIsWakeupEcu		EcucBooleanParamDef
BSW Description		
True: FrSM shall perform a wakeup for this cluster. False: FrSM shall never perform a wakeup for this FlexRay cluster.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrSM_00109]	

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMMainFunctionCycleTime		EcucFloatParamDef
BSW Description		
This parameter defines the cycle time in seconds of the periodic calling of FrSM main function.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrSM_00115]	

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMMinNumberOfColdstarter		EcucIntegerParamDef
BSW Description		
This parameter defines the number of coldstarter that should not be underrun. If this parameter is not configured the mainfunction shall not check the number of startup frames.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrSM_00168]

BSW Module	BSW Context
FrSM	FrSM/FrSMConfig/FrSMCluster
BSW Parameter	BSW Type
FrSMNumWakeupPatterns	EcucIntegerParamDef
BSW Description	
Maximum number of Wakeup Patterns the node may send before going to FRSM_STARTUP.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrSM_00165]

BSW Module	BSW Context
FrSM	FrSM/FrSMConfig/FrSMCluster
BSW Parameter	BSW Type
FrSMStartupRepetitions	EcucIntegerParamDef
BSW Description	
The number of times an ECU may repeat the startup procedure for a FlexRay cluster.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrSM_00069]

BSW Module	BSW Context
FrSM	FrSM/FrSMConfig/FrSMCluster
BSW Parameter	BSW Type
FrSMStartupRepetitionsWithWakeup	EcucIntegerParamDef
BSW Description	
The number of times an ECU may repeat the startup procedure including a wakeup for a FlexRay cluster.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrSM_00094]

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		BSW Type
FrSMTrcvStdbbyDelay		EcucFloatParamDef
BSW Description		
<p>The duration of timer $t_{TrcvStdbbyDelay}$ in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (FrlfGdCycle).</p> <p>A value of 0 shall imply that the timer is not used.</p>		
Template Description		
<p>The duration of timer $t_{TrcvStdbbyDelay}$ in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value.</p> <p>Not specifying a value or a value of 0 shall imply that the timer is not used.</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.transceiverStandbyDelay		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrSM_00170]

BSW Module	BSW Context	
FrSM	FrSM	
BSW Parameter		BSW Type
FrSMGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration parameters of the FlexRay State Manager.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrSM_00107]

BSW Module	BSW Context	
FrSM	FrSM/FrSMGeneral	
BSW Parameter		BSW Type
FrSMAllSlotsSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrSM support to enable/disable the switching from key-slot/single-slot mode to all-slot mode.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrSM_00172]

BSW Module		BSW Context	
FrSM		FrSM/FrSMGeneral	
BSW Parameter		BSW Type	
FrSMDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00066]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMGeneral	
BSW Parameter		BSW Type	
FrSMSyncLossErrorIndicationName		EcucFunctionNameDef	
BSW Description			
Name of <Cdd>_SyncLossErrorIndication function that shall be called on loss of synchronization. If this parameter is omitted no indication shall take place.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00167]	

BSW Module		BSW Context	
FrSM		FrSM/FrSMGeneral	
BSW Parameter		BSW Type	
FrSMVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables and disables the version info API			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_FrSM_00108]	

C.5 Lin

C.5.1 Lin Driver Mapping

BSW Module	BSW Context	
Lin	Lin	
BSW Parameter	BSW Type	
LinDemEventParameterRefs	EcucParamConfContainerDef	
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Lin_00188]	

BSW Module	BSW Context	
Lin	Lin/LinDemEventParameterRefs	
BSW Parameter	BSW Type	
LIN_E_TIMEOUT	EcucReferenceDef	
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Timeout caused by hardware error" has occurred. If the reference is not configured the error shall be reported as DET error.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Lin_00189]	

BSW Module	BSW Context	
Lin	Lin	
BSW Parameter	BSW Type	
LinGeneral	EcucParamConfContainerDef	
BSW Description		
This container contains the parameters related to each LIN Driver Unit.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	

Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00183]

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter	BSW Type	
LinDevErrorDetect	EcucBooleanParamDef	
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Lin_00066]	

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter	BSW Type	
LinEcucPartitionRef	EcucReferenceDef	
BSW Description		
Maps the Lin driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Lin driver will operate as an independent instance in each of the partitions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Lin_00192]	

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter	BSW Type	
LinIndex	EcucIntegerParamDef	
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Lin_00179]
-------	------------------

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		BSW Type
LinTimeoutDuration		EcucIntegerParamDef
BSW Description		
Specifies the maximum number of loops for blocking function until a timeout is raised in short term wait loops		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Lin_00093]

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		BSW Type
LinVersionInfoApi		EcucBooleanParamDef
BSW Description		
Switches the Lin_GetVersionInfo function ON or OFF.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Lin_00067]

BSW Module	BSW Context	
Lin	Lin	
BSW Parameter		BSW Type
LinGlobalConfig		EcucParamConfContainerDef
BSW Description		
This container contains the global configuration parameter of the Lin driver.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Lin_00184]

BSW Module	BSW Context

Lin	Lin/LinGlobalConfig	
BSW Parameter		BSW Type
LinChannel	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration (parameters) of the LIN Controller(s).		
Template Description		
<p>@RESTRICT_TO_STANDARD:CP! A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one physical channel. Bus systems like CAN and LIN only have exactly one PhysicalChannel. A FlexRay cluster may have more than one PhysicalChannels that may be used in parallel for redundant communication.</p> <p>An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.# @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! This element represents a physical connection (in case of CAN, FlexRay, LIN) or a logical connection (VLAN in case of Ethernet) between communicating devices. @END_RESTRICT_TO_STANDARD!</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel		
Mapping Rule		Mapping Type
A LinChannel container is constructed per CommunicationConnector belonging to the CommunicationController associated with the owning Lin Module container		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Lin_00069]

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		BSW Type
LinChannelBaudRate	EcucIntegerParamDef	
BSW Description		
Specifies the baud rate of the LIN channel		
Template Description		
Channels speed in bits/s.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Lin_00180]

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		BSW Type
LinChannelEcuMWakeupSource	EcucReferenceDef	
BSW Description		
This parameter contains a reference to the Wakeup Source for this controller as defined in the ECU State Manager.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00185]

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	BSW Type
LinChannelEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps one single Lin channel to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Lin driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00193]

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	BSW Type
LinChannelId	EcucIntegerParamDef
BSW Description	
Identifies the LIN channel. Replaces LIN_CHANNEL_INDEX_NAME from the LIN SWS.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Implicit from each CommunicationConnector on the ECU representing a LIN channel. Increase the LinChannelId for each LIN channel created on the same CommunicationController, for each CommunicationController start indexing at zero.	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00181]

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	BSW Type
LinChannelWakeupSupport	EcucBooleanParamDef
BSW Description	
Specifies if the LIN hardware channel supports wake up functionality	
Template Description	

Defines whether the ECU shall be woken up by this CommunicationController.	
TRUE: wake up is possible	
FALSE: wake up is not supported	
Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationController.wakeUpByControllerSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00182]

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	BSW Type
LinClockRef	EcucReferenceDef
BSW Description	
Reference to the LIN clock source configuration, which is set in the MCU driver configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Lin_00094]

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	BSW Type
LinNodeType	EcucEnumerationParamDef
BSW Description	
Specifies the LIN node type of this channel.	
Template Description	
LinMaster: Describing the properties of the referring ecu as a LIN master.	
LinSlave: Describing the properties of the referring ecu as a LIN slave.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave	
Mapping Rule	Mapping Type
Set to MASTER if the CommunicationController is defined as a LinMaster. Set to SLAVE if the CommunicationController is defined as a LinSlave. In the System Template the LinMaster/LinSlave is connected to the LinChannel via a CommunicationConnector.	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_Lin_00191]
-------	------------------

C.5.2 Lin Interface Mapping

BSW Module	BSW Context	
LinIf	LinIf	
BSW Parameter	BSW Type	
LinIfGeneral	EcucParamConfContainerDef	
BSW Description		
This container contains the general parameters of LIN Interface module.		
Template Description		
LIN specific attributes		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinCluster		
Mapping Rule		Mapping Type
Container shall be created if the ECU is connected to a LIN Cluster.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00019]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter	BSW Type	
LinIfBusMirroringSupported	EcucBooleanParamDef	
BSW Description		
States if Bus Mirroring is enabled in the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if the Bus Mirroring is not used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00657]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter	BSW Type	
LinIfDevErrorDetect	EcucBooleanParamDef	
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID

valid	[ECUC_LinIf_00010]
-------	--------------------

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter	BSW Type	
LinIfMultipleDriversSupported	EcucBooleanParamDef	
BSW Description		
States if multiple drivers are supported by the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if multiple drivers are not used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00024]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter	BSW Type	
LinIfMultipleTrcvDriverSupported	EcucBooleanParamDef	
BSW Description		
States if multiple transceiver drivers are supported by the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if multiple transceiver drivers are not used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00025]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter	BSW Type	
LinIfNcOptionalRequestSupported	EcucBooleanParamDef	
BSW Description		
States if the node configuration commands Assign NAD and Conditional Change NAD are supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00026]

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfPublicCddHeaderFile		EcucStringParamDef	
BSW Description			
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00631]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfResponseErrorSignalChangedCallout		EcucFunctionNameDef	
BSW Description			
This parameter contains the name of the callout function that is called after a response error signal change. Only applicable for LIN slave nodes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00656]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfSaveConfigurationCallout		EcucFunctionNameDef	
BSW Description			
This parameter contains the name of the callout function that is called when a save configuration node configuration command is processed by this slave node. The service is only supported when this parameter is configured. Only applicable for LIN slave nodes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00651]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfTpSupported		EcucBooleanParamDef	
BSW Description			
States if the TP is included in the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if the TP is not used.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00045]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfTrcvDriverSupported		EcucBooleanParamDef	
BSW Description			
States if transceiver driver support is included in the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if transceiver drivers are not used.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00635]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGeneral	
BSW Parameter		BSW Type	
LinIfVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Switches the LinIf_GetVersionInfo function ON or OFF.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00053]	

BSW Module		BSW Context	
LinIf		LinIf	
BSW Parameter		BSW Type	

LinIfGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains the global configuration parameters of the LinIf.	
Template Description	
LIN specific attributes	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinCluster	
Mapping Rule	Mapping Type
Container shall be created if the ECU is connected to a LIN Cluster.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00020]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig
BSW Parameter	BSW Type
LinIfChannel	EcucParamConfContainerDef
BSW Description	
Describes each LIN channel the LinIf is connected to.	
Template Description	
The connection between the referencing ECU and the referenced channel via the referenced controller.	
Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController.	
Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector	
Mapping Rule	Mapping Type
Container shall be created if the CommunicationConnector belonging to the ECU is connected to a LinChannel.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00364]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfBusIdleTimeoutPeriod	EcucFloatParamDef
BSW Description	
Bus idle timeout in seconds. According to the LIN protocol specification, the bus idle timeout period shall be in range [4, 10] seconds.	
Template Description	
This attribute shall be used to set an idle timeout period for the enclosing LinPhysicalChannel.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinPhysicalChannel.busIdleTimeoutPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00655]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfCddRef	EcucForeignReferenceDef	
BSW Description	Reference to the CDD module description. This parameter is only required when LinIfWakeupConfirmationUL, LinIfScheduleRequestConfirmationUL, LinIfGotoSleepConfirmationUL and/or LinIfGotoSleepIndicationUL is set to CDD.	
Template Description	<p>Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure.</p> <p>As part of the BSW module description, the EcucModuleConfigurationValues element has two different roles:</p> <p>The recommendedConfiguration contains parameter values recommended by the BSW module vendor.</p> <p>The preconfiguredConfiguration contains values for those parameters which are fixed by the implementation and cannot be changed.</p> <p>These two EcucModuleConfigurationValues are used when the base EcucModuleConfigurationValues (as part of the base ECU configuration) is created to fill parameters with initial values.</p>	
M2 Parameter	ECUCDescriptionTemplate::EcucModuleConfigurationValues	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00637]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfChannelRef	EcucReferenceDef	
BSW Description	Reference to the channel definition in the LIN driver.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00003]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfComMNetworkHandleRef	EcucReferenceDef	
BSW Description	Unique handle to identify one LIN network. Reference to one of the network handles configured for the ComM.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00626]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfFrame	EcucParamConfContainerDef
BSW Description	
Generic container for all types of LIN frames.	
Template Description	
LIN specific attributes to the FrameTriggering	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering	
Mapping Rule	Mapping Type
Create container for each LinFrameTriggering aggregated by the PhysicalChannel representing the regarded LIN channel.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00367]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	BSW Type
LinIfChecksumType	EcucEnumerationParamDef
BSW Description	
Type of checksum that the frame is using.	
This parameter is optional because in case of sporadic frames it should not be set.	
Template Description	
Type of checksum that the frame is using. This attribute is optional because in case of sporadic frames it should not be set.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.linChecksum	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00005]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	BSW Type
LinIfFixedFrameSdu	EcucParamConfContainerDef
BSW Description	
In case this is a fixed frame this is the SDU (response). This container represents an eight byte array. The Byte order shall be MSB first. Only applicable to LIN master nodes.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00012]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu
BSW Parameter	BSW Type
LinIfFixedFrameSduByte	EcucParamConfContainerDef
BSW Description	
This container represents a byte within the 8 byte array.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00013]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu/LinIfFixedFrameSduByte
BSW Parameter	BSW Type
LinIfFixedFrameSduBytePos	EcucIntegerParamDef
BSW Description	
Index of the Byte in the SDU (response) 8 byte array.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00014]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu/LinIfFixedFrameSduByte
BSW Parameter	BSW Type
LinIfFixedFrameSduByteVal	EcucIntegerParamDef
BSW Description	
Byte value in the SDU (response) 8-byte array.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00015]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		BSW Type
LinIfFrameId		EcucIntegerParamDef
BSW Description		
ID of the LIN frame. The Protected ID including parity is calculated by the generation tool.		
Template Description		
To describe a frames identifier on the communication system, usually with a fixed identifierValue. For LinSporadicFrames the attribute shall be ignored.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.identifier		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00638]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		BSW Type
LinIfFrameIndex		EcucIntegerParamDef
BSW Description		
PID index of the frame. This index is used in the AssignFrameIdentifierRange node configuration service to identify the frame(s) to which a new PID shall be assigned. It corresponds to the order of the frames in the configurable frames list in the node attributes section of the LDF / NCF of the slave node. Only relevant for LIN slave nodes.		
Template Description		
This attribute is used to order the elements and allows an assignment of Pids to ConfigurableFrames that are defined in the slave.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinOrderedConfigurableFrame.index		
Mapping Rule		Mapping Type
1:1 mapping. Only relevant if CommunicationController is defined as a LinSlave.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00653]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		BSW Type
LinIfFrameType		EcucEnumerationParamDef
BSW Description		

<p>This parameter defines the type of frame (e.g. sporadic frame). For master nodes, all frame types are permitted. A sporadic slot may be used by a set of unconditional frames in the role of substitution frames. For slave nodes, only following types are permitted: Unconditional, MRF, SRF, Event-triggered. An event-triggered slot may be used by a set of unconditional frames in the role of substitution frames.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
see details in EnumerationLiteralDef descriptions	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00017]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	BSW Type
ASSIGN	EcucEnumerationLiteralDef
BSW Description	
AssignFrameId	
Template Description	
Schedule entry for an Assign Frame Id master request.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::AssignFrameId	
Mapping Rule	Mapping Type
Use FrameType "Assign" if ScheduleEntry is an "AssignFrameId".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	BSW Type
ASSIGN_FRAME_ID_RANGE	EcucEnumerationLiteralDef
BSW Description	
AssignFrameIdRange	
Template Description	
AssignFrameIdRange generates an assign frame PID range request.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::AssignFrameIdRange	
Mapping Rule	Mapping Type
Use FrameType "Assign_Frame_Id_Range" if ScheduleEntry is an "Assign FrameIdRange".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	BSW Type
ASSIGN_NAD	EcucEnumerationLiteralDef
BSW Description	

AssignNAD	
Template Description	
Schedule entry for an Assign NAD master request.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::AssignNad	
Mapping Rule	Mapping Type
Use FrameType "Assign_NAD" if ScheduleEntry is an "AssignNad".	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	BSW Type
CONDITIONAL	EcucEnumerationLiteralDef
BSW Description	
Conditional Change NAD	
Template Description	
Generates an conditional change NAD request. See ISO 17987 protocol specification for more information.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ConditionalChangeNad	
Mapping Rule	Mapping Type
Use FrameType "CONDITIONAL" if ScheduleEntry is an "ConditionalChangeNad".	
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	BSW Type
EVENT_TRIGGERED	EcucEnumerationLiteralDef
BSW Description	
Event triggered frame	
Template Description	
<p>An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response.</p> <p>The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable.</p> <p>The event controlled frame shall not contain any Pdus.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinEventTriggeredFrame	
Mapping Rule	Mapping Type
Derive the type from System Description	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
-------------------	--------------------

LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type
FREE		EcucEnumerationLiteralDef
BSW Description		
FreeFormat		
Template Description		
Representing freely defined data.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::FreeFormat		
Mapping Rule		Mapping Type
Use FrameType "Free Format" if ScheduleEntry is a "FreeFormatEntry".		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type
MRF		EcucEnumerationLiteralDef
BSW Description		
Master Request Frame		
Template Description		
<p>The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent.</p> <p>For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::FrameTriggering		
Mapping Rule		Mapping Type
Use common Frame for Master Request.		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type
SAVE_CONFIGURATION		EcucEnumerationLiteralDef
BSW Description		
SaveConfiguration		
Template Description		
This service is used to notify a slave node to store its configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::SaveConfigurationEntry		
Mapping Rule		Mapping Type
Use FrameType "Save_Configuration" is an "SaveConfiguration".		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	

BSW Parameter		BSW Type
SPORADIC		EcucEnumerationLiteralDef
BSW Description		
Sporadic slot		
Template Description		
A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinSporadicFrame		
Mapping Rule		Mapping Type
Derive the type from System Description		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type
SRF		EcucEnumerationLiteralDef
BSW Description		
Slave Response Frame		
Template Description		
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent.		
For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::FrameTriggering		
Mapping Rule		Mapping Type
Use common Frame for Slave Response.		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type
UNASSIGN		EcucEnumerationLiteralDef
BSW Description		
UnassignFrameId		
Template Description		
Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::UnassignFrameId		
Mapping Rule		Mapping Type
Use FrameType "Unassign" if ScheduleEntry is an "UnassignFrameId".		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module		BSW Context	
LinIf		LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType	
BSW Parameter		BSW Type	
UNCONDITIONAL		EcucEnumerationLiteralDef	
BSW Description			
Unconditional Frame			
Template Description			
Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinUnconditionalFrame			
Mapping Rule			Mapping Type
Derive the type from System Description			full
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
LinIf		LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		BSW Type	
LinIfPduDirection		EcucChoiceContainerDef	
BSW Description			
Direction of the frame			
Template Description			
LIN specific attributes to the FrameTriggering			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering			
Mapping Rule			Mapping Type
Create container for each existing LinFrame.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00027]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter		BSW Type	
LinIfInternalPdu		EcucParamConfContainerDef	
BSW Description			
Represents a Diagnostic or Configuration frame : no Message ID (no PduId). Only applicable to LIN master nodes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_LinIf_00021]	

BSW Module		BSW Context	
LinIf		LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter		BSW Type	

LinIfRxPdu	EcucParamConfContainerDef
BSW Description	
represents a received PDU/frame	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.framePort	
Mapping Rule	Mapping Type
Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "in" FramePort	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00035]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu
BSW Parameter	BSW Type
LinIfRxIndicationUL	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter LinIfUserRxIndicationUL.	
If LinIfUserRxIndicationUL equals PDUR, the name of the <User_RxIndication> is fixed.	
If LinIfUserRxIndicationUL equals CDD, the name of the <User_RxIndication> is selectable.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00055]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu
BSW Parameter	BSW Type
LinIfRxPduRef	EcucReferenceDef
BSW Description	
Reference to the PDU that is received in this frame.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00036]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu
BSW Parameter	BSW Type

LinIfUserRxIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the indication of the successfully received LinIfRxPdu has to be routed via <User_RxIndication>.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00610]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection
BSW Parameter	BSW Type
LinIfSlaveToSlavePdu	EcucParamConfContainerDef
BSW Description	
Represents a slave-to-slave PDU/frame. Master does only send the header but doesn't receive the response. Only relevant for master nodes.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00040]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection
BSW Parameter	BSW Type
LinIfTxPdu	EcucParamConfContainerDef
BSW Description	
represents a transmitted PDU/frame	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.framePort	
Mapping Rule	
Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "out" FramePort	Mapping Type
	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00049]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu
BSW Parameter	BSW Type
LinIfTxConfirmationUL	EcucFunctionNameDef

BSW Description	
This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter LinIfUserTxUL. If LinIfUserTxUL equals PDUR, the name of the <User_TxConfirmation> is fixed. If LinIfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00054]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu
BSW Parameter	BSW Type
LinIfTxPduId	EcucIntegerParamDef
BSW Description	
Identifier of the Pdu for the upper layer.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00050]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu
BSW Parameter	BSW Type
LinIfTxPduRef	EcucReferenceDef
BSW Description	
Reference to the PDU that is transmitted in this frame.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00051]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu
BSW Parameter	BSW Type
LinIfTxTriggerTransmitUL	EcucFunctionNameDef
BSW Description	

<p>This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter LinIfUserTxUL. If LinIfUserTxUL equals PDUR, the name of the <User_TriggerTransmit> is fixed. If LinIfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00628]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu
BSW Parameter	BSW Type
LinIfUserTxUL	EcucEnumerationParamDef
BSW Description	
<p>This parameter defines the upper layer (UL) module to which the trigger of the transmitted LinTxPdu (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted LinTxPdu has to be routed (via the <User_TxConfirmation>).</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00609]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	BSW Type
LinIfSubstitutionFrames	EcucParamConfContainerDef
BSW Description	
<p>List of sporadic frames that can be sent in a sporadic frame slot (master node) or list of unconditional frames that can be sent in an event-triggered frame slot (slave node).</p>	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinSporadicFrame.substitutedFrame	
Mapping Rule	Mapping Type
emulate reference from System Description	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00042]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfSubstitutionFrames
BSW Parameter	BSW Type
LinIfFramePriority	EcucIntegerParamDef

BSW Description	
Priority of sporadic frame in a master node or of event-triggered frame in slave node.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinSporadicFrame.substitutedFrame	
Mapping Rule	Mapping Type
In the System Description the priority is described by the Order of the UnconditionalFrames	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00513]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfSubstitutionFrames
BSW Parameter	BSW Type
LinIfSubstitutionFrameRef	EcucReferenceDef
BSW Description	
Reference to an unconditional Frame that is used as sporadic frame in a master node or event-triggered frame in a slave node.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00041]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfGotoSleepConfirmationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the confirmation of the goto-sleep command shall be sent.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00601]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfGotoSleepIndicationUL	EcucEnumerationParamDef
BSW Description	

This parameter defines the upper layer (UL) module to which the indication of the goto-sleep command shall be sent. Only used for LIN Slave nodes, ignored for master nodes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00652]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Defines the interval of calls to main functions per channel in seconds.	
Template Description	
Time base is mandatory for the master. It is not used for slaves.	
LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time."	
The time base shall be specified AUTOSAR conform in seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster.timeBase	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00639]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfMaxFrameCnt	EcucIntegerParamDef
BSW Description	
Maximum number of Frames. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00636]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel

BSW Parameter		BSW Type
LinIfNodeType		EcucChoiceContainerDef
BSW Description		
This container defines the LIN node type of this channel.		
Template Description		
LinMaster: Describing the properties of the referring ecu as a LIN master.		
LinSlave: Describing the properties of the referring ecu as a LIN slave.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave		
Mapping Rule		Mapping Type
Set to MASTER if the CommunicationController is defined as a LinMaster. Set to SLAVE if the CommunicationController is defined as a LinSlave. In the System Template the LinMaster/LinSlave is connected to the LinChannel via a CommunicationConnector.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00654]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType	
BSW Parameter		BSW Type
LinIfMaster		EcucParamConfContainerDef
BSW Description		
Each Master can only be connected to one physical channel. This could be compared to the Node parameter in a LDF file.		
Template Description		
Describing the properties of the referring ecu as a LIN master.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster		
Mapping Rule		Mapping Type
Create container if the regarded ECU contains a CommunicationController that is defined as a LinMaster. In the System Template the LinMaster is connected to the LinChannel via a CommunicationConnector.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00512]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfMaster	
BSW Parameter		BSW Type
LinIfJitter		EcucFloatParamDef
BSW Description		
The jitter specifies the differences between the maximum and minimum delay from time base tick to the header sending start point in seconds.		
Template Description		

The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves.

LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)."

The jitter shall be specified AUTOSAR conform in seconds.

M2 Parameter

SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster.timeBaseJitter

Mapping Rule

1:1 mapping

Mapping Type

full

Mapping Status

valid

ECUC Parameter ID

[ECUC_LinIf_00629]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType
BSW Parameter	BSW Type
LinIfSlave	EcucParamConfContainerDef
BSW Description	
Describes all parameters which are only relevant for a LIN Slave node.	
Template Description	
Describing the properties of the referring ecu as a LIN slave.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00649]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave
BSW Parameter	BSW Type
LinIfLinProtocolVersion	EcucEnumerationParamDef
BSW Description	
Defines the LIN protocol version of the slave node. This information is relevant for the LIN conformance test execution.	
Template Description	
<p>LinSlaveConfig.protocolVersion: Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different.</p> <p>LinCommunicationController.protocolVersion: Version specifier for a communication protocol.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.protocolVersion, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinCommunicationController.protocolVersion	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00647]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave	
BSW Parameter		BSW Type
LinIfNodeConfigurationIdentification		EcucParamConfContainerDef
BSW Description		
This container is mandatory for all LIN 2.x and ISO17987 LIN slave nodes, and ignored for LIN 1.3 slave nodes and all master nodes,		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00650]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification	
BSW Parameter		BSW Type
LinIfConfiguredNAD		EcucIntegerParamDef
BSW Description		
Slave node configured NAD.		
Template Description		
LinSlave.configuredNad: To distinguish LIN slaves that are used twice or more within the same cluster.		
LinSlaveConfig.configuredNad: To distinguish LIN slaves that are used twice or more within the same cluster.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.configuredNad		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00643]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification	
BSW Parameter		BSW Type
LinIfFunctionId		EcucIntegerParamDef
BSW Description		
LIN function Id.		
Template Description		
LinSlave.functionId: LIN function ID		
LinSlaveConfig.functionId: LIN function ID.		
M2 Parameter		

SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.functionId, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.functionId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00646]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	BSW Type
LinIfInitialNAD	EcucIntegerParamDef
BSW Description	
Slave node initial NAD.	
Template Description	
LinSlave.initialNad: This attribute represents the initial NAD.	
LinSlaveConfig.initialNad: Initial NAD of the LIN slave.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.initialNad, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.initialNad	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00642]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	BSW Type
LinIfNasTimeout	EcucFloatParamDef
BSW Description	
N_As timeout in seconds.	
Template Description	
Value of the N_AS timeout. Unit: seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.nasTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00644]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	BSW Type
LinIfSupplierId	EcucIntegerParamDef
BSW Description	
LIN consortium or ISO LIN supplier Id.	

Template Description	
LinSlave.supplierId: LIN Supplier ID	
LinSlaveConfig.supplierId: LIN Supplier ID.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.supplierId, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.supplierId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00645]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	BSW Type
LinIfVariantId	EcucIntegerParamDef
BSW Description	
LIN variant Id.	
Template Description	
LinSlave.variantId: Specifies the Variant ID	
LinSlaveConfig.variantId: Specifies the Variant ID.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.variantId, SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlaveConfig.variantId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00641]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave
BSW Parameter	BSW Type
LinIfResponseErrorSignal	EcucReferenceDef
BSW Description	
Reference to the response_error signal. Mandatory for all LIN 2.x and ISO LIN slave nodes, not relevant for LIN 1.3 slave nodes.	
Template Description	
This ISignal shall be taken to transport the responseError bit.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinErrorResponse.responseError	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00648]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfScheduleChangeNextTimeBase	EcucBooleanParamDef	
BSW Description		
Enables/disables the switch to a new schedule table at the start of the next time base after status check. True: LinIf selects a new schedule table in next main function. Only applicable for LIN Master nodes.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00640]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfScheduleRequestConfirmationUL	EcucEnumerationParamDef	
BSW Description		
This parameter defines the upper layer (UL) module to which the confirmation of the successfully performed schedule table change shall be sent. Only applicable to LIN master nodes.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinIf_00600]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfScheduleTable	EcucParamConfContainerDef	
BSW Description		
Describes a schedule table. Each LinIfChannel may have several schedule tables. Each schedule table can only be connected to one channel. Mandatory for LIN Master nodes. The SHORT-NAME of the LinIfScheduleTable container represents the symbolic name of the schedule table.		
Template Description		
The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable		
Mapping Rule		Mapping Type

Create container for each ScheduleTable that is defined for this channel.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00365]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter	BSW Type	
LinIfEntry	EcucParamConfContainerDef	
BSW Description		
Describes an entry in the schedule table (also known as Frame Slot).		
Template Description		
Table entry in a LinScheduleTable. Specifies what will be done in the frame slot.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry		
Mapping Rule		Mapping Type
Each RelativelyScheduledTiming element in the System Description requires the creation of a LinIfEntry. RelativelyScheduledTiming.scheduleTable decides to which schedule table the LinIfEntry belongs.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00366]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	BSW Type	
LinIfCollisionResolvingRef	EcucReferenceDef	
BSW Description		
Reference to the schedule table, which resolves the collision. This parameter is only used if the referenced frames are event triggered frames.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinEventTriggeredFrame.collisionResolvingSchedule		
Mapping Rule		Mapping Type
Emulate the reference from the System Description.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00007]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	BSW Type	
LinIfDelay	EcucFloatParamDef	
BSW Description		
Delay to next entry in schedule table in seconds.		
Template Description		
Relative delay between this tableEntry and the start of the successor in the schedule table in seconds.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry.delay		
Mapping Rule		Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00009]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	BSW Type	
LinIfEntryIndex	EcucIntegerParamDef	
BSW Description		
Position of the Frame Entry in the Schedule Table. The first entry index in the schedule table is 0.		
Template Description		
Relative position in the schedule table. The first entry index in the schedule table is 0.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry.positionInTable		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00011]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	BSW Type	
LinIfFrameRef	EcucReferenceDef	
BSW Description		
Reference to the frames that belong to this schedule table entry.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ApplicationEntry.frameTriggering		
Mapping Rule		Mapping Type
Emulate reference from the System Description		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00016]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter	BSW Type	
LinIfResumePosition	EcucEnumerationParamDef	
BSW Description		
Defines where a RUN_CONTINUOUS schedule table shall proceed in case it has been interrupted by a RUN_ONCE table.		
Template Description		
Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.resumePosition		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00033]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter	BSW Type	
LinIfRunMode	EcucEnumerationParamDef	
BSW Description	The schedule table can be executed in two different modes.	
Template Description	The schedule table can be executed in two different modes.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00034]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter	BSW Type	
LinIfScheduleTableIndex	EcucIntegerParamDef	
BSW Description	This is the unique index used by upper layers to identify a schedule. Note that the NULL_SCHEDULE for each channel must have index 0.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00037]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter	BSW Type	
LinIfTransceiverDrvConfig	EcucParamConfContainerDef	
BSW Description	This container contains the configuration parameters of each underlying LIN Transceiver Driver.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinIf_00046]	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfTransceiverDrvConfig	
BSW Parameter	BSW Type	
LinIfTrcvIdRef	EcucReferenceDef	

BSW Description	
Logical handle of the underlying LIN transceiver to be served by the LIN Interface.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00047]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	BSW Type
LinIfWakeupConfirmationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the confirmation of the wake-up shall be sent.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinIf_00602]

C.5.3 LinTp Mapping

BSW Module	BSW Context
LinTp	LinTp
BSW Parameter	BSW Type
LinTpGeneral	EcucParamConfContainerDef
BSW Description	
Container that holds all LIN transport protocol general parameters.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00617]

BSW Module	BSW Context
LinTp	LinTp/LinTpGeneral
BSW Parameter	BSW Type
LinTpChangeParameterApi	EcucBooleanParamDef
BSW Description	
This parameter, if set to true, enables the LinTp_ChangeParameter Api for this Module.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00638]

BSW Module	BSW Context
LinTp	LinTp/LinTpGeneral
BSW Parameter	BSW Type
LinTpVersionInfoApi	EcucBooleanParamDef
BSW Description	
Switches the LinTp_GetVersionInfo function ON or OFF.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00068]

BSW Module	BSW Context
LinTp	LinTp
BSW Parameter	BSW Type
LinTpGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains the global configuration parameters of the LinTp.	
Template Description	
TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.	
M2 Parameter	
SystemTemplate::TransportProtocols::LinTpNode	
Mapping Rule	Mapping Type
Create container if the regarded ECU is a LinTpNode.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00056]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	BSW Type
LinTpChannelConfig	EcucParamConfContainerDef
BSW Description	
This container contains the channel specific configuration parameters of LinTp.	
Template Description	
M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu	

Mapping Rule		Mapping Type
Create container for each NSdu that is received by the regarded ECU.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00071]	

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig	
BSW Parameter		BSW Type
LinTpChannelRef		EcucReferenceDef
BSW Description		
Index of the channel this LinTp channel belongs to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00073]	

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig	
BSW Parameter		BSW Type
LinTpDropNotRequestedNad		EcucBooleanParamDef
BSW Description		
Configures if TP Frames of not requested LIN-Slaves are dropped or not.		
TRUE: Drop TP Frames of not requested LIN-Slaves FALSE: Keep TP Frames of not requested LIN-Slaves		
Only used for LIN Master nodes, ignored for slave nodes.		
Template Description		
Configures if TP Frames of not requested LIN-Slaves are dropped or not.		
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.dropNotRequestedNad		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00072]	

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig	
BSW Parameter		BSW Type
LinTpMaxNumberOfRespPendingFrames		EcucIntegerParamDef
BSW Description		
Configures the maximum number of allowed response pending frames. Only used for LIN Master nodes, ignored for slave nodes.		
Template Description		
Configures the maximum number of allowed response pending frames.		
M2 Parameter		

SystemTemplate::TransportProtocols::LinTpConnection.maxNumberOfRespPendingFrames	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00624]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	BSW Type
LinTpP2Max	EcucFloatParamDef
BSW Description	
P2*max timeout when a response pending frame is expected in seconds. Note that the minimum value of LinTpP2Max shall be more than or equal to the value of LinTpP2Timing. Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.	
M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.p2Max	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00622]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	BSW Type
LinTpP2Timing	EcucFloatParamDef
BSW Description	
Definition of the P2max timeout observation parameter in seconds. Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
P2 timeout observation parameter.	
M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.p2Timing	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00625]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	BSW Type
LinTpScheduleChangeDiag	EcucBooleanParamDef
BSW Description	

Enables or disables the call of BswM_LinTp_RequestMode() to diagnostic request/response schedule.	
false: BswM is not called true: BswM is called	
Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00070]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	BSW Type
LinTpMaxRxNSduCnt	EcucIntegerParamDef
BSW Description	
Maximum number of NSdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00635]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	BSW Type
LinTpMaxTxNSduCnt	EcucIntegerParamDef
BSW Description	
Maximum number of NSdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00636]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig

BSW Parameter		BSW Type
LinTpRxNSdu		EcucParamConfContainerDef
BSW Description		
This container exists once for each received N-SDU on any channel the node is connected to. This N-SDU produces meta data items of type LIN_NAD_8.		
Template Description		
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu		
Mapping Rule		Mapping Type
Create container for each NSdu that is received by the regarded ECU.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinTp_00428]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		BSW Type
LinTpNcr		EcucFloatParamDef
BSW Description		
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.		
Template Description		
This attribute defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.		
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.timeoutCr		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinTp_00632]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		BSW Type
LinTpRxNSduChannelRef		EcucReferenceDef
BSW Description		
Index of the channel this N-SDU belongs to.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_LinTp_00060]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		BSW Type
LinTpRxNSduld		EcucIntegerParamDef

BSW Description	
The identifier of the Transport Protocol message. This ID will be used by upper layers to call LinTp_ChangeParameter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00061]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		BSW Type
LinTpRxNSduNad		EcucIntegerParamDef
BSW Description		
A N-SDU transported on LIN is identified using the NAD for the specific slave.		
Template Description		
To distinguish LIN slaves that are used twice or more within the same cluster.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad		
Mapping Rule	Mapping Type	
Find connection from NSdu to CommunicationController	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00062]	

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		BSW Type
LinTpRxNSduPduRef		EcucReferenceDef
BSW Description		
Reference to the global PDU		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00063]	

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		BSW Type
LinTpTxNSdu		EcucParamConfContainerDef
BSW Description		
This container exists once for each transmitted N-SDU on any channel the node is connected to. This N-SDU consumes meta data items of type LIN_NAD_8.		
Template Description		

M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu	
Mapping Rule	Mapping Type
Create container for each NSdu that is received by the regarded ECU.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00511]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpMaxBufReq	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of times the LinTp should request upper layer for the Tx Buffer. It is also used to limit the number of retries for PduR_LinTpCopyTxData when no timer is active.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00637]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpNas	EcucFloatParamDef
BSW Description	
Value in seconds of the N_As timeout. N_As is the time for transmission of a LIN frame (any N_PDU) on the part of the sender.	
Template Description	
Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.	
M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.timeoutAs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00633]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpNcs	EcucFloatParamDef
BSW Description	
Value in seconds of the performance requirement of N-Cs. N-Cs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.	
Template Description	
The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.	

M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.timeoutCs	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00634]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpTxNSduChannelRef	EcucReferenceDef
BSW Description	
Index of the channel this N-SDU belongs to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00064]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpTxNSdulId	EcucIntegerParamDef
BSW Description	
The identifier of the Transport Protocol message. This ID will be the one that is communicated with upper layers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00065]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	BSW Type
LinTpTxNSduNad	EcucIntegerParamDef
BSW Description	
A N-SDU transported on LIN is identified using the NAD for the specific slave.	
Template Description	
To distinguish LIN slaves that are used twice or more within the same cluster.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad	
Mapping Rule	Mapping Type
Find connection from NSdu to CommunicationController	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_LinTp_00066]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter	BSW Type	
LinTpTxNSduPduRef	EcucReferenceDef	
BSW Description	Reference to the global PDU	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_LinTp_00067]	

C.6 Ethernet

C.6.1 Ethernet Driver Mapping

BSW Module	BSW Context	
Eth	Eth	
BSW Parameter	BSW Type	
EthConfigSet	EcucParamConfContainerDef	
BSW Description	This container contains the configuration parameters and sub containers of the AUTOSAR Eth module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00015]	

BSW Module	BSW Context	
Eth	Eth/EthConfigSet	
BSW Parameter	BSW Type	
EthCtrlConfig	EcucParamConfContainerDef	
BSW Description	Configuration of the individual controller	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00006]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlConfigEgress	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet controler egress behavior.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00046]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress
BSW Parameter	BSW Type
EthCtrlConfigEgressFifo	EcucParamConfContainerDef
BSW Description	
Represents a Fifo at the egress side.	
Template Description	
Defines a Fifo for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortFifo	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00047]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressFifo
BSW Parameter	BSW Type
EthCtrlConfigEgressFifoBufLenByte	EcucIntegerParamDef
BSW Description	
Length of Fifo elements in bytes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00051]

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigEgressFifoBufTotal		EcucIntegerParamDef	
BSW Description			
Fifo buffer count.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00050]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigEgressFifoldx		EcucIntegerParamDef	
BSW Description			
Egress Fifo index.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00048]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigEgressFifoPriorityAssignment		EcucIntegerParamDef	
BSW Description			
Message egress prority assignment.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00049]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress	
BSW Parameter		BSW Type	
EthCtrlConfigEgressLastSchedulerRef		EcucReferenceDef	
BSW Description			

Reference to the scheduler which is the last in the egress structure.	
Template Description	
Defines which CouplingPortScheduler is the last in the egress port structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortDetails.lastEgressScheduler	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00052]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress
BSW Parameter	BSW Type
EthCtrlConfigScheduler	EcucParamConfContainerDef
BSW Description	
Represents a Scheduler on the egress side.	
Template Description	
Defines a scheduler for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00053]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler
BSW Parameter	BSW Type
EthCtrlConfigSchedulerPredecessor	EcucParamConfContainerDef
BSW Description	
Defines an ordered list of predecessors for this scheduler.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00054]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/ EthCtrlConfigSchedulerPredecessor
BSW Parameter	BSW Type
EthCtrlConfigSchedulerPredecessorOrder	EcucIntegerParamDef
BSW Description	
Defines the order of the scheduler predecessors.	
Template Description	

Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor	
Mapping Rule	Mapping Type
Defined by the order of CouplingPortScheduler.predecessor.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00055]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/ EthCtrlConfigSchedulerPredecessor
BSW Parameter	BSW Type
EthCtrlConfigSchedulerPredecessorRef	EcucChoiceReferenceDef
BSW Description	
Choice reference to the scheduler predecessor.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00056]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress
BSW Parameter	BSW Type
EthCtrlConfigShaper	EcucParamConfContainerDef
BSW Description	
Represents a Shaper an the egress side.	
Template Description	
Defines a shaper for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00057]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigShaper
BSW Parameter	BSW Type
EthCtrlConfigShaperIdleSlope	EcucIntegerParamDef
BSW Description	
Defines the increase of credit in bits per second for the AVB shaper.	
Template Description	
Defines the increase of credit in bits per second for the AVB shaper.	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper.idleSlope	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00058]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigShaper
BSW Parameter	BSW Type
EthCtrlConfigShaperPredecessorFifoRef	EcucReferenceDef
BSW Description	
Reference to the fifo which is the predecessor for this shaper.	
Template Description	
Defines the CouplingPortFifo which provides the input to this shaper.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper.predecessorFifo	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00059]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlConfigIngress	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet controller ingress behavior.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00040]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigIngress
BSW Parameter	BSW Type
EthCtrlConfigIngressFifo	EcucParamConfContainerDef
BSW Description	
Represents a Fifo at the ingress side.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00041]

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigIngress/EthCtrlConfigIngressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigIngressFifoBufLenByte		EcucIntegerParamDef	
BSW Description			
Length of Fifo elements in bytes.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00045]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigIngress/EthCtrlConfigIngressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigIngressFifoBufTotal		EcucIntegerParamDef	
BSW Description			
Fifo buffer count.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00044]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigIngress/EthCtrlConfigIngressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigIngressFifoldx		EcucIntegerParamDef	
BSW Description			
Ingress Fifo index.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00043]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigIngress/EthCtrlConfigIngressFifo	
BSW Parameter		BSW Type	
EthCtrlConfigIngressFifoPriorityAssignment		EcucIntegerParamDef	
BSW Description			

Message ingress priority assignment.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00042]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the Ethernet controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00065]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlEnableMii	EcucBooleanParamDef
BSW Description	
Enables / Disables Media Independent Interface (MII) for transceiver access	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00012]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlEnableRxInterrupt	EcucBooleanParamDef
BSW Description	
Enables / Disables receive interrupt	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00010]	

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		BSW Type
EthCtrlEnableTxInterrupt		EcucBooleanParamDef
BSW Description		
Enables / Disables transmit interrupt		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00011]	

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		BSW Type
EthCtrlIdx		EcucIntegerParamDef
BSW Description		
Specifies the instance ID of the configured controller.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00007]	

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		BSW Type
EthCtrlMacLayerSpeed		EcucEnumerationParamDef
BSW Description		
Defines the baud rate of the MAC layer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00063]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		BSW Type	
EthCtrlMacLayerSubType		EcucEnumerationParamDef	
BSW Description			
Defines the MAC layer subtype of a switch port			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00062]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		BSW Type	
EthCtrlMacLayerType		EcucEnumerationParamDef	
BSW Description			
Defines the MAC layer type of the ethernet controller.			
Template Description			
Specifies the mac layer type of the CouplingPort.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.macLayerType			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Eth_00039]	

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType	
BSW Parameter		BSW Type	
ETH_MAC_LAYER_TYPE_XGMII		EcucEnumerationLiteralDef	
BSW Description			
MAC layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)			
Template Description			
Mac layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xGMII			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
Eth		Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType	
BSW Parameter		BSW Type	
ETH_MAC_LAYER_TYPE_XMII		EcucEnumerationLiteralDef	
BSW Description			

MAC layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
Template Description	
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType
BSW Parameter	BSW Type
ETH_MAC_LAYER_TYPE_XXGMII	EcucEnumerationLiteralDef
BSW Description	
MAC layer interface (data) bandwidth class 10Gbit/s	
Template Description	
Mac layer interface (data) bandwidth class 10Gbit/s	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xXGMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthCtrlPhyAddress	EcucStringParamDef
BSW Description	
Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order.	
Regular Expression: [0-9a-fA-F]{2}[:-][0-9a-fA-F]{2}{5}	
Template Description	
Media Access Control address (MAC address) that uniquely identifies each EthernetCommunicationController in the network.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationController.macUnicastAddress	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00020]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	BSW Type
EthDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00016]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_ACCESS	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Controller access failed" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00017]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_ALIGNMENT	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Alignment Error" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00026]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_CRC	EcucReferenceDef
BSW Description	

Reference to the DemEventParameter which shall be issued when the error "CRC Failure" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00023]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_LATECOLLISION	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Late Collisions" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00029]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_MULTIPLECOLLISION	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Multiple Collisions" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00028]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_OVERSIZEFRAME	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Oversized Frame" has occurred.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00025]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_RX_FRAMES_LOST	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "receive frames lost" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00021]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_SINGLECOLLISION	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Single Collisions" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00027]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs
BSW Parameter	BSW Type
ETH_E_UNDERSIZEFRAME	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Undersized Frame" has occurred.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00024]

BSW Module	BSW Context
Eth	Eth
BSW Parameter	BSW Type
EthGeneral	EcucParamConfContainerDef
BSW Description	
General configuration of Ethernet Driver module	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00001]

BSW Module	BSW Context
Eth	Eth/EthGeneral
BSW Parameter	BSW Type
EthCtrlOffloading	EcucParamConfContainerDef
BSW Description	
Configuration of hardware offloading features.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00030]

BSW Module	BSW Context
Eth	Eth/EthGeneral/EthCtrlOffloading
BSW Parameter	BSW Type
EthCtrlEnableOffloadChecksumICMP	EcucBooleanParamDef
BSW Description	
Enables / Disables hardware offloading for ICMP checksums.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_Eth_00032]
-------	------------------

BSW Module	BSW Context	
Eth	Eth/EthGeneral/EthCtrlOffloading	
BSW Parameter	BSW Type	
EthCtrlEnableOffloadChecksumIPv4	EcucBooleanParamDef	
BSW Description		
Enables / Disables hardware offloading for IPv4 checksums.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00031]

BSW Module	BSW Context	
Eth	Eth/EthGeneral/EthCtrlOffloading	
BSW Parameter	BSW Type	
EthCtrlEnableOffloadChecksumTCP	EcucBooleanParamDef	
BSW Description		
Enables / Disables hardware offloading for TCP checksums.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00033]

BSW Module	BSW Context	
Eth	Eth/EthGeneral/EthCtrlOffloading	
BSW Parameter	BSW Type	
EthCtrlEnableOffloadChecksumUDP	EcucBooleanParamDef	
BSW Description		
Enables / Disables hardware offloading for UDP checksums.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00034]

BSW Module	BSW Context
Eth	Eth/EthGeneral

BSW Parameter		BSW Type
EthDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00003]

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		BSW Type
EthEcucPartitionRef		EcucReferenceDef
BSW Description		
Maps the Ethernet driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet driver will operate as an independent instance in each of the partitions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00064]

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		BSW Type
EthGetDropCountApi		EcucBooleanParamDef
BSW Description		
Enables / Disables Eth_GetCounterValues API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Eth_00035]

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		BSW Type
EthGetEtherStatsApi		EcucBooleanParamDef

BSW Description	
Enables / Disables Eth_GetEtherStats API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00036]

BSW Module	BSW Context
Eth	Eth/EthGeneral
BSW Parameter	BSW Type
EthGetTxErrorCounterValuesApi	EcucBooleanParamDef
BSW Description	
Enables/Disables Eth_GetTxErrorCounterValues API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00061]

BSW Module	BSW Context
Eth	Eth/EthGeneral
BSW Parameter	BSW Type
EthGetTxStatsApi	EcucBooleanParamDef
BSW Description	
Enables/Disables Eth_GetTxStats API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00060]

BSW Module	BSW Context
Eth	Eth/EthGeneral
BSW Parameter	BSW Type
EthGlobalTimeSupport	EcucBooleanParamDef
BSW Description	
Enables/Disables the GlobalTime APIs used amongst others by Global Time Synchronization over Ethernet.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00037]

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter	BSW Type	
EthIndex	EcucIntegerParamDef	
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00018]	

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter	BSW Type	
EthMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
Specifies the period of main function Eth_MainFunction in seconds. Ethernet driver does not require this information but the BSW scheduler.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Eth_00022]	

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter	BSW Type	
EthMaxCtrlsSupported	EcucIntegerParamDef	
BSW Description		
Limits the total number of supported controllers.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00002]

BSW Module	BSW Context
Eth	Eth/EthGeneral
BSW Parameter	BSW Type
EthVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables / Disables version info API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Eth_00004]

C.6.2 Ethernet Interface Mapping

BSW Module	BSW Context
EthIf	EthIf
BSW Parameter	BSW Type
EthIfConfigSet	EcucParamConfContainerDef
BSW Description	
Collecting container for all parameters with post-build configuration classes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00010]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet
BSW Parameter	BSW Type
EthIfController	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of EthIfController.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00025]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfCtrlIdx	EcucIntegerParamDef	
BSW Description	This parameter provides a zero-based consecutive index of the Ethernet Communication Controllers. Upper layer BSW modules and the EthIf itself use this index to identify a Ethernet CC.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00026]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfCtrlMtu	EcucIntegerParamDef	
BSW Description	Specifies the maximum transmission unit (MTU) of the EthIfCtrl in [bytes].	
	Note: In case a VLAN tag is used for the EthIfCtrl, the frame length of the Ethernet frame will increase by 4 bytes.	
Template Description	This attribute specifies the maximum transmission unit in bytes.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.maximumTransmissionUnit	
Mapping Rule	Mapping Type	
Different MTU values may be defined for different VLANs. Therefore the maximumTransmissionUnit is specified in the EthernetCommunicationConnector. The value that is defined in the CommunicationConnector shall be used as the value of this parameter.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00032]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfEthTrcvRef	EcucReferenceDef	
BSW Description	Reference to an Ethernet transceiver, which is handled by the Ethernet Interface.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00028]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfMaxTxBufsTotal	EcucIntegerParamDef	
BSW Description	Limits the total number of transmit buffers.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00002]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfPhysControllerRef	EcucReferenceDef	
BSW Description	Reference to a physical Ethernet controller, which is handled by the Ethernet Interface.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00027]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter	BSW Type	
EthIfSwitchRefOrPortGroupRef	EcucChoiceReferenceDef	
BSW Description	<p>The choice reference allows to configure either the EthIfController references an EthIfSwitch or an EthIfSwitchPortGroup.</p> <p>Reference to a EthIfSwitchPortGroup. In case port groups are controlled by PNC EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_LINK_INFO. In case port groups are controlled by the EhtIfController EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_CONTROL.</p>	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00048]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		BSW Type
EthIfVlanId		EcucIntegerParamDef
BSW Description		
A virtual-LAN is identified by this attribute according to IEEE 802.1Q.		
Template Description		
A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanConfig.vlanIdentifier		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00029]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		BSW Type
EthIfFrameOwnerConfig		EcucParamConfContainerDef
BSW Description		
Configuration of Ethernet frame owner		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00011]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfFrameOwnerConfig	
BSW Parameter		BSW Type
EthIfFrameType		EcucIntegerParamDef
BSW Description		
Selects the Ethernet frame type.		
Template Description		
Ethernet specific attributes to the Frame.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame::AbstractEthernetFrame		
Mapping Rule		Mapping Type
If an AbstractEthernetFrame is defined in the System Extract then it may be possible to derive this parameter from this information.		partial
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00012]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfFrameOwnerConfig	
BSW Parameter		BSW Type

EthIfOwner	EcucIntegerParamDef
BSW Description	
Selects the owner of an Ethernet frame type. The owner is a zero based index into the callback function configuration 'EthIfRxIndicationConfig'. I.e. an Ethernet frame of type IPv4 (0x800) at index 0 will call the first callback function configured in 'EthIfRxIndicationConfig'.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00013]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet
BSW Parameter	BSW Type
EthIfPhysController	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of EthIfPhysController.	
The usage of EthIfEthCtrlRef and EthIfWEthCtrlRef is exclusive OR.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00045]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfPhysController
BSW Parameter	BSW Type
EthIfEthCtrlRef	EcucReferenceDef
BSW Description	
Reference to a physical Ethernet controller, which is handled by a specific Ethernet controller driver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00047]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfPhysController
BSW Parameter	BSW Type
EthIfPhysControllerIdx	EcucIntegerParamDef

BSW Description	
This parameter provides a zero-based consecutive index of the physical Ethernet controllers. Upper layer BSW modules and the Ethernet Interface itself use this index to identify a physical Ethernet controller.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00046]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfPhysController
BSW Parameter	BSW Type
EthIfPhysCtrlRxMainFunctionPriorityProcessing	EcucParamConfContainerDef
BSW Description	
Configuration of ingress FIFO based main function processing.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00050]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfPhysController/EthIfPhysCtrlRxMainFunctionPriorityProcessing
BSW Parameter	BSW Type
EthIfPhysCtrlRxIndicationIterations	EcucIntegerParamDef
BSW Description	
Max number of Ethernet frames polled per main function invocation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00052]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfPhysController/EthIfPhysCtrlRxMainFunctionPriorityProcessing
BSW Parameter	BSW Type
EthIfPhysCtrlRxIngressFifoRef	EcucReferenceDef
BSW Description	

Reference to the reception FIFO.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00053]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet/EthlfPhysController/EthlfPhysCtrlRxMainFunctionPriority Processing
BSW Parameter	BSW Type
EthlfPhysCtrlRxMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Specifies the period of main function in seconds.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00051]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet/EthlfPhysController
BSW Parameter	BSW Type
EthlfWEthCtrlRef	EcucReferenceDef
BSW Description	
Reference to a physical Wireless Ethernet controller, which is handled by a specific Wireless Ethernet controller driver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00073]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet
BSW Parameter	BSW Type
EthlfRxIndicationConfig	EcucParamConfContainerDef
BSW Description	
Configuration of receive callback functions.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00014]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfRxIndicationConfig
BSW Parameter	BSW Type
EthIfRxIndicationFunction	EcucFunctionNameDef
BSW Description	
Specifies receive indication callback function.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00015]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet
BSW Parameter	BSW Type
EthIfSwitch	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of EthIfSwitches.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00036]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfSwitch
BSW Parameter	BSW Type
EthIfSwitchIdx	EcucIntegerParamDef
BSW Description	
This parameter provides a zero-based consecutive index of the Ethernet Interface Switches. Upper layer BSW modules and the EthIf itself use this index to identify a Ethernet Switch.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00037]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfSwitch	
BSW Parameter	BSW Type	
EthIfSwitchRef	EcucReferenceDef	
BSW Description		
Reference to a Ethernet Switch, which is handled by a specific Ethernet Switch driver.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00038]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter	BSW Type	
EthIfSwitchMgmtInfoIndicationConfig	EcucParamConfContainerDef	
BSW Description		
Configuration of Switch Management callback function.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00065]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfSwitchMgmtInfoIndicationConfig	
BSW Parameter	BSW Type	
EthIfSwitchMgmtInfoIndicationFunction	EcucFunctionNameDef	
BSW Description		
Enables/Disables the ingress Switch management info indication redirected call to upper layers who registered for the call.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthIf_00067]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter	BSW Type	
EthIfSwitchPortGroup	EcucParamConfContainerDef	
BSW Description		
<p>This container contains the configuration of EthIfSwitchPortGroups.</p> <p>If EthIfSwitchPortGroups are controlled by PNC one EthIfSwitchPortGroup per PNC shall exist.</p> <p>The host port shall be part of all EthIfSwitchPortGroups.</p> <p>The up link port of a master switch and the up link port of the slave switch shall be part of all EthIfSwitchPortGroups that contain EthSwtPorts belonging to the slave switch.</p>		
Template Description		
<p>EcucInstance.ethSwitchPortGroupDerivation: Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcucInstance. If not defined the derivation shall not be done.</p> <p>CouplingPort.pncMapping: Reference to the partial networks this CouplingPort participates in.</p>		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcucInstance.ethSwitchPortGroupDerivation, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.pncMapping		
Mapping Rule		Mapping Type
<p>Derive EthIfSwitchPortGroup from M2 according to the following requirements only if EcucInstance.ethSwitchPortGroupDerivation is defined and set to TRUE:</p> <ul style="list-style-type: none"> * For each EthernetPhysicalChannel that has CouplingPorts connected (CouplingPort has a VlanMembership referring to the EthernetPhysicalChannel) exactly one EthIfSwitchPortGroup shall be derived containing all connected CouplingPorts via the EthIfPortRef. Thus a EthSwtPort may be part in several EthIfSwitchPortGroups. * For each PNC that is referenced by at least one CouplingPort exactly one EthIfSwitchPortGroup shall be derived. The referenced CouplingPort shall be part of the EthIfSwitchPortGroup via the EthIfPortRef. Thus a EthSwtPort may be part in several EthIfSwitchPortGroups. * If the CouplingPorts have no reference to any PNC or all referenced PNCs have no relation to this EthernetPhysicalChannel then the derived EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_CONTROL, because this EthIfSwitchPortGroup is switched by EthSM. * If the CouplingPorts have at least one reference to any PNC that has a relation to this EthernetPhysicalChannel then the derived EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_LINK_INFO, because this EthIfSwitchPortGroup is only used for link status accumulation towards EthSM. 		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00057]

BSW Module	BSW Context
------------	-------------

EthIf	EthIf/EthIfConfigSet/EthIfSwitchPortGroup	
BSW Parameter		BSW Type
EthIfPortRef	EcucReferenceDef	
BSW Description		
Reference to an Ethernet Switch Port.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00060]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfSwitchPortGroup	
BSW Parameter		BSW Type
EthIfSwitchPortGroupIdx	EcucIntegerParamDef	
BSW Description		
This parameter provides a zero-based consecutive index of the Ethernet Switch Port Groups. Upper layer BSW modules and the EthIf itself use this index to identify an Ethernet Switch Port Group.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00058]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfSwitchPortGroup	
BSW Parameter		BSW Type
EthIfSwitchPortGroupRefSemantics	EcucEnumerationParamDef	
BSW Description		
Defines how the EthIfSwitchRefOrPortGroupRef referring to a EthIfSwitchPortGroup shall be interpreted.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00059]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		BSW Type
EthIfSwitchTimeStampIndicationConfig	EcucParamConfContainerDef	

BSW Description	
Configuration of Switch timestamp indications.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Ethlf_00066]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet/EthlfSwitchTimeStampIndicationConfig
BSW Parameter	BSW Type
EthlfSwitchEgressTimeStampIndicationFunction	EcucFunctionNameDef
BSW Description	
Enables/Disables to upper layers an egress timestamp indication function.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Ethlf_00068]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet/EthlfSwitchTimeStampIndicationConfig
BSW Parameter	BSW Type
EthlfSwitchIngressTimeStampIndicationFunction	EcucFunctionNameDef
BSW Description	
Enables/Disables to upper layers an ingress timestamp indication function.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_Ethlf_00069]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfConfigSet
BSW Parameter	BSW Type
EthlfTransceiver	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of EthlfTransceiver.	
The usage of EthlfEthTrcvRef and EthlfWEthTrcvRefis exclusive OR.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00042]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfTransceiver
BSW Parameter	BSW Type
EthIfEthTrcvRef	EcucReferenceDef
BSW Description	
Reference to an Ethernet transceiver, which is handled by a specific Ethernet transceiver driver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00044]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfTransceiver
BSW Parameter	BSW Type
EthIfTransceiverIdx	EcucIntegerParamDef
BSW Description	
This parameter provides a zero-based consecutive index of the Ethernet transceivers. Upper layer BSW modules and the Ethernet Interface itself use this index to identify an Ethernet transceiver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00043]

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfTransceiver
BSW Parameter	BSW Type
EthIfWEthTrcvRef	EcucReferenceDef
BSW Description	
Reference to an Wireless Ethernet transceiver, which is handled by a specific Wireless Ethernet transceiver driver.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00074]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfConfigSet	
BSW Parameter		BSW Type
EthlfTrcvLinkStateChgConfig		EcucParamConfContainerDef
BSW Description		
Specifies link state change callback function		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00018]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfConfigSet/EthlfTrcvLinkStateChgConfig	
BSW Parameter		BSW Type
EthlfTrcvLinkStateChgFunction		EcucFunctionNameDef
BSW Description		
Specifies link state change callback function		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00019]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfConfigSet	
BSW Parameter		BSW Type
EthlfTxConfirmationConfig		EcucParamConfContainerDef
BSW Description		
Configuration of transmit indication callback functions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00016]	

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfTxConfirmationConfig	
BSW Parameter		BSW Type
EthIfTxConfirmationFunction		EcucFunctionNameDef
BSW Description		
Specifies transmit indication callback function		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00017]

BSW Module	BSW Context	
EthIf	EthIf	
BSW Parameter		BSW Type
EthIfGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration parameters of the Ethernet Interface.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00001]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		BSW Type
EthIfDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00004]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		BSW Type

EthIfEnableRxInterrupt	EcucBooleanParamDef
BSW Description	
Enables / Disables receive interrupt.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_EthIf_00005]

BSW Module	BSW Context
EthIf	EthIf/EthIfGeneral
BSW Parameter	BSW Type
EthIfEnableSignalQualityApi	EcucBooleanParamDef
BSW Description	
Enable/disable the APIs read and clear the signal quality.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_EthIf_00076]

BSW Module	BSW Context
EthIf	EthIf/EthIfGeneral
BSW Parameter	BSW Type
EthIfEnableTxInterrupt	EcucBooleanParamDef
BSW Description	
Enables / Disables the transmit interrupt.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_EthIf_00006]

BSW Module	BSW Context
EthIf	EthIf/EthIfGeneral
BSW Parameter	BSW Type
EthIfEnableWEthApi	EcucBooleanParamDef
BSW Description	
Enables / Disables API's for WEth / WEthTrcv	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00075]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfGetAndResetMeasurementDataApi	EcucBooleanParamDef
BSW Description	
Enables / Disables the Get and Reset Measurement Data API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00072]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfGetBaudRate	EcucBooleanParamDef
BSW Description	
Enables / Disables GetBaudRate API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00034]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfGetCounterState	EcucBooleanParamDef
BSW Description	
Enables / Disables GetCounterState API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_Ethlf_00035]
-------	--------------------

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfGeneral	
BSW Parameter	BSW Type	
EthlfGetCtrlIdxList	EcucBooleanParamDef	
BSW Description	Enables / Disables GetCtrlIdxList API.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00070]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfGeneral	
BSW Parameter	BSW Type	
EthlfGetTransceiverWakeupModeApi	EcucBooleanParamDef	
BSW Description	Enables / Disables Ethlf_GetTransceiverWakeupMode API	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00041]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfGeneral	
BSW Parameter	BSW Type	
EthlfGetVlanIdSupport	EcucBooleanParamDef	
BSW Description	Enables / Disables GetVlanId API.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Ethlf_00071]	

BSW Module	BSW Context	
Ethlf	Ethlf/EthlfGeneral	

BSW Parameter		BSW Type
EthIfGlobalTimeSupport		EcucBooleanParamDef
BSW Description		
Enables/Disables the Global Time APIs used amongst others by Global Time Synchronization over Ethernet.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00039]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		BSW Type
EthIfMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Specifies the period of main function EthIf_MainFunctionRx and EthIf_MainFunctionTx in seconds. Ethernet Interface does not require this information but the BSW scheduler.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00023]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		BSW Type
EthIfMainFunctionStatePeriod		EcucFloatParamDef
BSW Description		
Specifies the period of main function EthIf_MainFunctionState in seconds. Ethernet Interface does not require this information but the BSW scheduler.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthIf_00056]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		BSW Type
EthIfMaxTrcvsTotal		EcucIntegerParamDef

BSW Description	
Limits the total number of transceivers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00003]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfPortStartupActiveTime	EcucFloatParamDef
BSW Description	
Denote the time delay after the mode "ETH_MODE_ACTIVE" of all EthlfSwitchPorts are requested via Ethlf_StartAllPorts.	
This is only used for ports in EthlfSwtpPortGroups which are not referenced by any EthlfController.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00055]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfPublicCddHeaderFile	EcucStringParamDef
BSW Description	
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00024]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfRxIndicationIterations	EcucIntegerParamDef

BSW Description	
Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within Ethlf_MainFunctionRx.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00030]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfSetForwardingModeApi	EcucBooleanParamDef
BSW Description	
Enables /disables Ethlf_SetForwardingMode API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00062]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfSignalQualityCheckPeriod	EcucFloatParamDef
BSW Description	
Specifies the period in units of seconds in which the signal quality it polled in the context of Ethlf_MainfunctionState. The value shall be an integral multiple of EthlfMainFunctionStatePeriod.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00077]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfStartAutoNegotiation	EcucBooleanParamDef
BSW Description	
Enables / Disables StartAutoNegotiation API.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00033]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfSwitchManagementSupport	EcucBooleanParamDef
BSW Description	
Enables/Disables the Switch management APIs to support a Switch-port specific communication attribute access.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00064]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfSwitchOffPortTimeDelay	EcucFloatParamDef
BSW Description	
Denote the time delay after the mode "ETH_MODE_DOWN" of a EthlfSwitchPortGroup will be executed.	
This is only used for EthlfSwtpPortGroups which are not referenced by any EthlfController.	
The time delay shall be greater than the UdpNm timings, because UdpNm shall finish its shutdown handling. (Repeat Message State, Prepare Bus-Sleep state, Bus-Sleep state).	
Template Description	
Switch off delay for CouplingPorts in seconds. It denotes the delay of switching off couplingPorts after the request to switch off a couplingPort was issued. (e.g. switch off of Ethernet switch ports).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCluster.couplingPortSwitchoff Delay	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00054]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfTrcvLinkStateChgMainReload	EcucIntegerParamDef

BSW Description	
Specifies the frequency of transceiver link state change checks in each period of main function Ethlf_MainFunctionTx.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00009]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfVerifyConfigApi	EcucBooleanParamDef
BSW Description	
Enables /disables Ethlf_VerifyConfig API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00063]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables / Disables version info API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Ethlf_00007]

BSW Module	BSW Context
Ethlf	Ethlf/EthlfGeneral
BSW Parameter	BSW Type
EthlfVersionInfoApiMacro	EcucBooleanParamDef
BSW Description	
Enables / Disables version info API macro implementation.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00008]

BSW Module	BSW Context
EthIf	EthIf/EthIfGeneral
BSW Parameter	BSW Type
EthIfWakeUpSupport	EcucBooleanParamDef
BSW Description	
Configures if wakeup is supported or not.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthIf_00040]

C.6.3 Ethernet Switch Driver Mapping

BSW Module	BSW Context
EthSwt	EthSwt
BSW Parameter	BSW Type
EthSwtConfig	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet Switch.	
Template Description	
A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingElement	
Mapping Rule	Mapping Type
For each CouplingElement with couplingType=switch one EthSwtConfig shall be created.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00001]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	BSW Type
EthSwtArITableEntryTimeout	EcucFloatParamDef
BSW Description	
If present, this parameter specifies the timeout in seconds for removing unused entries from the ARL table of the Ethernet switch. Otherwise, entries are not removed automatically.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00127]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig
BSW Parameter	BSW Type
EthSwTClockSynchronizationSupport	EcucBooleanParamDef
BSW Description	
<p>This parameter defines, if a Ethernet switch shall enable clock synchronization with another Ethernet switch to which it is connected via uplink port.</p> <p>If this parameter is set to TRUE the clock synchronization between connected Ethernet switches is activated and the clocks of the Ethernet switches are synchronized. If this parameter is set to FALSE the clock synchronization between connected Ethernet switches is deactivated.</p> <p>This parameter shall only be set to TRUE if the Ethernet switch hardware supports clock synchronization.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00128]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig
BSW Parameter	BSW Type
EthSwTConfigEcucPartitionRef	EcucReferenceDef
BSW Description	
<p>Maps the configuration of one single Ethernet switch to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet switch driver is mapped to.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00130]

BSW Module	BSW Context

EthSwt	EthSwt/EthSwtConfig
BSW Parameter	BSW Type
EthSwtDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00016]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtDemEventParameterRefs
BSW Parameter	BSW Type
ETHSWT_E_ACCESS	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Ethernet Switch Access Failure" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00006]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtDemEventParameterRefs
BSW Parameter	BSW Type
ETHSWT_E_SYNCPORT2PHY	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Ethernet switch port and the referenced Ethernet transceiver are in contradicting modes" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00125]

BSW Module	BSW Context
-------------------	--------------------

EthSw	EthSw/EthSwConfig
BSW Parameter	
EthSwDropDoubleTagged	EcucBooleanParamDef
BSW Description	
<p>This parameter defines if a switch shall drop double tagged (Q in Q) frames.</p> <p>If this parameter is set to TRUE double tagged frames are dropped at all ports.</p> <p>If this parameter is set to FALSE, then double tagged frames are forwarded. If double tagging is used as a feature, this parameter must be set to FALSE.</p> <p>This parameter shall only be set to TRUE when Switch-HW supports the filtering of double tagged frames as filtering by SW is NOT possible!</p>	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthSw_00073]

BSW Module	BSW Context
EthSw	EthSw/EthSwConfig
BSW Parameter	
EthSwIdx	EcucIntegerParamDef
BSW Description	
Specifies the instance ID of the configured Ethernet Switch.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthSw_00004]

BSW Module	BSW Context
EthSw	EthSw/EthSwConfig
BSW Parameter	
EthSwManagementEthCtrlRef	EcucReferenceDef
BSW Description	
Reference to the Ethernet controller connected to the management port where the management frames will be transmitted/received.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthSw_00110]

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig	
BSW Parameter		BSW Type	
EthSwTManagementPortRef		EcucReferenceDef	
BSW Description			
Reference to the port where the management CPU is connected to.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00111]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig	
BSW Parameter		BSW Type	
EthSwTNvm		EcucParamConfContainerDef	
BSW Description			
Configuration of one Ethernet Switch Nvm usage in case the module requires non volatile memory in the Ecu to store switch configuration.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00043]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig/EthSwTNvm	
BSW Parameter		BSW Type	
EthSwTConfigurationNvmBlockDescriptorRef		EcucReferenceDef	
BSW Description			
Reference to the Nvm block description in the Nvm module configuration to store e.g. the port mirror configurations			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00134]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig/EthSwTNvm	
BSW Parameter		BSW Type	

EthSwtTableNvmBlockDescriptorRef	EcucReferenceDef
BSW Description	
Reference to the Nvm block description in the Nvm module configuration to store e.g. the learned ARL table	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00044]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	BSW Type
EthSwtPort	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet Switch Port.	
Template Description	
A CouplingPort is used to connect a CouplingElement with an EcucInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort	
Mapping Rule	
For each CouplingElement.couplingPort of a CouplingElement with coupling Type=switch one EthSwtPort shall be created.	
Mapping Type	
full	
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00005]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	BSW Type
EthSwtPortEgress	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet Switch Port Egress behavior.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
local	
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00007]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	BSW Type
EthSwtPortEgressLastSchedulerRef	EcucReferenceDef

BSW Description	
Reference to the port scheduler which is the last in the egress port structure.	
Template Description	
Defines which CouplingPortScheduler is the last in the egress port structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortDetails.lastEgressScheduler	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00008]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress
BSW Parameter	BSW Type
EthSwTPortFifo	EcucParamConfContainerDef
BSW Description	
Represents a Fifo in the egress port.	
Template Description	
Defines a Fifo for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortFifo	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00011]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortFifo
BSW Parameter	BSW Type
EthSwTPortEgressFifoldx	EcucIntegerParamDef
BSW Description	
Specifies the instance ID of the fifo of the configured Ethernet switch egress port	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00132]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortFifo
BSW Parameter	BSW Type
EthSwTPortFifoMinimumLength	EcucIntegerParamDef
BSW Description	
FIFO minimum length in Byte. This assignment is used to configure a guaranteed size of a configured FIFO.	
Template Description	

FIFO minimum length in Byte. An actual configuration/hardware may use a bigger value.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortFifo.minimumFifoLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00098]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortFifo
BSW Parameter	BSW Type
EthSwTPortFifoTrafficClassAssignment	EcucIntegerParamDef
BSW Description	
Defines which traffic classes are assigned to this Fifo.	
Template Description	
Defines a set of Traffic Classes which shall be handled by this Fifo.	
range: 0-7	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortFifo.assignedTrafficClass	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00012]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress
BSW Parameter	BSW Type
EthSwTPortScheduler	EcucParamConfContainerDef
BSW Description	
Represents a Scheduler in the egress port.	
Template Description	
Defines a scheduler for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00017]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortScheduler
BSW Parameter	BSW Type
EthSwTPortSchedulerAlgorithm	EcucEnumerationParamDef
BSW Description	
Defines the scheduler algorithm.	
Template Description	
Defines the schedule algorithm to be used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.portScheduler	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00018]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortScheduler/EthSwtPortSchedulerAlgorithm	
BSW Parameter		BSW Type
ETHSWT_SCHEDULER_DEFICIT_ROUND_ROBIN		EcucEnumerationLiteralDef
BSW Description		
deficit round robin		
Template Description		
Schedule algorithm "deficit round robin"		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCouplingPortSchedulerEnum.deficitRoundRobin		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortScheduler/EthSwtPortSchedulerAlgorithm	
BSW Parameter		BSW Type
ETHSWT_SCHEDULER_STRICT_PRIORITY		EcucEnumerationLiteralDef
BSW Description		
strict priority		
Template Description		
Schedule algorithm "strict priority"		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCouplingPortSchedulerEnum.strictPriority		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortScheduler/EthSwtPortSchedulerAlgorithm	
BSW Parameter		BSW Type
ETHSWT_SCHEDULER_WEIGHTED_ROUND_ROBIN		EcucEnumerationLiteralDef
BSW Description		
weighted round robin		
Template Description		
Schedule algorithm "weighted round robin"		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCouplingPortSchedulerEnum.weightedRoundRobin		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortScheduler	
BSW Parameter		BSW Type
EthSwTPortSchedulerPredecessor		EcucParamConfContainerDef
BSW Description		
Defines an ordered list of predecessors for this scheduler.		
Template Description		
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00019]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortScheduler/EthSwTPortSchedulerPredecessor	
BSW Parameter		BSW Type
EthSwTPortEgressPredecessorRef		EcucChoiceReferenceDef
BSW Description		
Choice reference to the scheduler predecessor.		
Template Description		
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00010]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortScheduler/EthSwTPortSchedulerPredecessor	
BSW Parameter		BSW Type
EthSwTPortSchedulerPredecessorOrder		EcucIntegerParamDef
BSW Description		
Defines the order of the scheduler predecessors.		
<p>This value has to be understood as a relative value, i.e. the value shows only the relative ordering of the elements. The highest value has the highest priority and gaps are allowed (not dense based). The values need to be unique within one EthSwTPortScheduler.</p>		
Template Description		

Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortScheduler.predecessor	
Mapping Rule	Mapping Type
Defined by the order of CouplingPortScheduler.predecessor.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00020]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress
BSW Parameter	BSW Type
EthSwTPortShaper	EcucParamConfContainerDef
BSW Description	
Represents a Shaper in the egress port.	
Template Description	
Defines a shaper for the CouplingPort egress structure.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00021]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortShaper
BSW Parameter	BSW Type
EthSwTPortEgressPredecessorFifoRef	EcucReferenceDef
BSW Description	
Reference to the fifo which is the predecessor for this shaper.	
Template Description	
Defines the CouplingPortFifo which provides the input to this shaper.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper.predecessorFifo	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00009]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortEgress/EthSwTPortShaper
BSW Parameter	BSW Type
EthSwTPortShaperIdleSlope	EcucIntegerParamDef
BSW Description	
Defines the increase of credit in bits per second for the AVB shaper.	
Template Description	
Defines the increase of credit in bits per second for the AVB shaper.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortShaper.idleSlope	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00042]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortIdx	EcucIntegerParamDef
BSW Description	
Specifies the instance ID of the configured Ethernet Switch Port.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00013]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortIngress	EcucParamConfContainerDef
BSW Description	
Configuration of one Ethernet Switch Port ingress behavior.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00014]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress
BSW Parameter	BSW Type
EthSwTPortIngressDefaultPriority	EcucIntegerParamDef
BSW Description	
Default priority for ingress.	
Template Description	
Standard output-priority outgoing Frames will be tagged with.	
<p>Defines the priority that received frames are assigned together with the VLAN Id (default-Vlan). The values from 0 (best effort) to 7 (highest) are allowed.</p> <p>In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.defaultPriority	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00096]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress	
BSW Parameter		BSW Type
EthSwTPortIngressDefaultVlan		EcucIntegerParamDef
BSW Description		
Default VLAN for ingress.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.defaultVlan		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00095]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress	
BSW Parameter		BSW Type
EthSwTPortIngressDropUntagged		EcucBooleanParamDef
BSW Description		
Defines the ingress behavior for untagged frames.		
Template Description		
Defines the handling of frames at the ingress port.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.receiveActivity		
Mapping Rule		Mapping Type
If CouplingPort.receiveActivity is set to "dropUntagged" then EthSwTPortIngress DropUntagged shall be set to true.		full
If CouplingPort.receiveActivity is set to something different than "dropUntagged" then EthSwTPortIngressDropUntagged shall be set to false.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00097]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress	
BSW Parameter		BSW Type
EthSwTPortIngressVlanModification		EcucIntegerParamDef
BSW Description		
If this parameter is defined all messages which arrive at this ingress port will be tagged with this VLAN Id. This tagging happen also if the arriving message already has a VLAN Id, it will be overwritten by the defined one.		
If this parameter is not defined no changes to the VLAN Id shall happen at this ingress port.		
Template Description		

All incoming messages at this CouplingPort shall be tagged with this VLAN Id. This tagging is performed regardless whether the message already has a VLAN tag or is untagged, an existing VLAN tag will be overwritten.

This feature is XOR with CoupligPort.defaultVlan.

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.vlanModifier	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00015]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress
BSW Parameter	BSW Type
EthSwTPortPolicer	EcucParamConfContainerDef
BSW Description	
Definition of Rate Policing parameters.	
Template Description	
Defines a rate policy on a CouplingPort.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00074]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPortPolicer
BSW Parameter	BSW Type
EthSwTPortRatePolicedByteCount	EcucIntegerParamDef
BSW Description	
Amount of Byte Counts (excluding Header information) which can be received in a configured EthSwTPortRatePolicedTimeInterval.	
Template Description	
Amount of data in bytes (excluding header information) that can be received to define the rate policy.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy.dataLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00075]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPortPolicer
BSW Parameter	BSW Type
EthSwTPortRatePolicedPriority	EcucIntegerParamDef
BSW Description	
Defines the priority which this rate policy shall be limited on. If no priority is given this rate policy is not considering priority.	

Template Description	
Defines the priority which this rate policy shall be limited on. If no priority is given this rate policy is not considering priority.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy.priority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00077]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPortPolicer
BSW Parameter	BSW Type
EthSwTPortRatePolicedTimeInterval	EcucFloatParamDef
BSW Description	
Time interval in seconds where a configured EthSwTPortRatePolicedByteCount can be received without a rate limitation.	
Template Description	
Time interval used to define the base of the rate policy.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy.timeInterval	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00076]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPortPolicer
BSW Parameter	BSW Type
EthSwTPortRateViolationAction	EcucEnumerationParamDef
BSW Description	
Action to be taken when the rate policy criteria defined for this EthSwTPortPolicer are met.	
Template Description	
Defines the action to be performed when this rate policy is violated.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy.policyAction	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00078]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPortPolicer/EthSwTPortRateViolationAction
BSW Parameter	BSW Type
BLOCK_SOURCE	EcucEnumerationLiteralDef
BSW Description	
All incoming traffic from the violating Source based on the MAC-Address is blocked.	
Template Description	

If the rate policy is violated the CouplingPort this CouplingPortRatePolicy is defined on shall block all frames from the MAC-Address the violation was caused by.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicyAction Enum.blockSource	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortPolicer/EthSwtPortRateViolationAction
BSW Parameter	BSW Type
DROP_FRAME	EcucEnumerationLiteralDef
BSW Description	
The received frame which led to the violation of the rate policy is dropped.	
Template Description	
If the rate policy is violated the frame shall be dropped.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicyAction Enum.dropFrame	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortPolicer
BSW Parameter	BSW Type
EthSwtPortRateVlanMembershipRef	EcucReferenceDef
BSW Description	
References the Vlans this rate policy shall apply to.	
If no EthSwtPortRateVlanMembershipRef is configured the rate policing applies only on the configured EthSwtPortRatePolicedPriority.	
Template Description	
Defines the VLANs this rate policy shall be limited on. If no VLAN is given this rate policy is not considering VLAN tags.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortRatePolicy.vLan	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00081]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress
BSW Parameter	BSW Type
EthSwtPortTrafficClassAssignment	EcucIntegerParamDef

BSW Description	
If this parameter is defined all arriving messages at this ingress port shall be assigned this traffic class.	
If this parameter is not defined no general port based traffic class assignment is done.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortDetails.ethernetTrafficClassAssignment	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00023]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress
BSW Parameter	BSW Type
EthSwtPriorityRegeneration	EcucParamConfContainerDef
BSW Description	
Defines a priority regeneration where the EthSwtPriorityRegenerationIngressPriority is replaced by EthSwtPriorityRegenerationRegeneratedPriority.	
The EthSwtPriorityRegeneration is optional in case no priority regeneration shall be performed.	
In case a EthSwtPriorityRegeneration is defined it shall have 8 mappings, one for each priority.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortDetails.ethernetPriorityRegeneration	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00057]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPriorityRegeneration
BSW Parameter	BSW Type
EthSwtPriorityRegenerationIngressPriority	EcucIntegerParamDef
BSW Description	
Message priority of the incoming message.	
Template Description	
Message priority of the incoming message.	
range: 0-7	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPriorityRegeneration.ingressPriority	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00058]

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPriorityRegeneration	
BSW Parameter	BSW Type	
EthSwTPriorityRegenerationRegeneratedPriority	EcucIntegerParamDef	
BSW Description		
Message priority the incoming message will be tagged with.		
Template Description		
Regenerated message priority.		
range: 0-7		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPriorityRegeneration.regeneratedPriority		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00059]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress	
BSW Parameter	BSW Type	
EthSwTPriorityTrafficClassAssignment	EcucParamConfContainerDef	
BSW Description		
Defines a priority based traffic class assignment. All messages with a specific priority (EthSwTPriorityTrafficClassAssignmentPriority) arriving at this ingress port or, if enabled regenerated priorities (EthSwTPriorityRegeneration), shall be assigned to a traffic class (EthSwTPriorityTrafficClassAssignmentTrafficClass).		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortDetails.ethernetTrafficClassAssignment		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00027]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPriorityTrafficClassAssignment	
BSW Parameter	BSW Type	
EthSwTPriorityTrafficClassAssignmentPriority	EcucIntegerParamDef	
BSW Description		
Message priority.		
Template Description		
Defines a priority which is mapped onto a Traffic Class.		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortTrafficClassAssignment.priority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00028]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortIngress/EthSwTPriorityTrafficClassAssignment
BSW Parameter	BSW Type
EthSwTPriorityTrafficClassAssignmentTrafficClass	EcucIntegerParamDef
BSW Description	
Traffic Class value.	
Template Description	
Defines the Traffic Class which is assigned.	
range: 0-7	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPortTrafficClassAssignment.trafficClass	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00029]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortMacLayerSpeed	EcucEnumerationParamDef
BSW Description	
Defines the baud rate of the MAC layer.	
Template Description	
Specifies the mac layer type of the CouplingPort.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.macLayerType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00114]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSpeed
BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_100M	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSpeed
BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_10G	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Mac layer interface (data) bandwith class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSpeed
BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_10M	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Mac layer interface (data) bandwith class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSpeed
BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_1G	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Mac layer interface (data) bandwith class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	
-------	--

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortMacLayerSpeed	
BSW Parameter	BSW Type	
ETH_MAC_LAYER_SPEED_2500M	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort	
BSW Parameter	BSW Type	
EthSwtPortMacLayerSubType	EcucEnumerationParamDef	
BSW Description		
Defines the MAC layer subtype of this EthSwtPort.		
Template Description		
Specifies the mac layer type of the CouplingPort.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.macLayerType		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00113]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortMacLayerSubType	
BSW Parameter	BSW Type	
REDUCED	EcucEnumerationLiteralDef	
BSW Description		
Reduced media-independent interface		
Template Description		
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortMacLayerSubType	

BSW Parameter		BSW Type	
REVERSED		EcucEnumerationLiteralDef	
BSW Description			
reversed media-independent interface (to provide direct connection between two Ethernet MACs)			
Template Description			
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSubType	
BSW Parameter		BSW Type	
SERIAL		EcucEnumerationLiteralDef	
BSW Description			
low-power and low pin-count serial 8b/10b-coded media-independent interface			
Template Description			
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSubType	
BSW Parameter		BSW Type	
STANDARD		EcucEnumerationLiteralDef	
BSW Description			
standard media-independent interface			
Template Description			
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTConfig/EthSwTPort/EthSwTPortMacLayerSubType	
BSW Parameter		BSW Type	
UNIVERSAL_SERIAL		EcucEnumerationLiteralDef	
BSW Description			
Universal low-power and low pin-count serial 8b/10b-coded media-independent interface			
Template Description			

Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortMacLayerType	EcucEnumerationParamDef
BSW Description	
Defines the MAC layer type of this EthSwTPort.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00072]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortPhysicalLayerType	EcucEnumerationParamDef
BSW Description	
Defines the physical layer type of this EthSwTPort.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00054]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTConfig/EthSwTPort
BSW Parameter	BSW Type
EthSwTPortPredefinedMacAddresses	EcucStringParamDef
BSW Description	
Specifies a list of 48-bit physical addresses (MAC addresses) which can be reached via this port in network byte order. Note that further addresses can be learned during runtime.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.macMulticastAddress	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00032]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort	
BSW Parameter		BSW Type
EthSwtPortRole		EcucEnumerationParamDef
BSW Description		
Set a special role of the Ethernet switch port. It is either a host port or a up link port. If not configured it is a standard port.		
Template Description		
Defines the role this CouplingPort takes in the context of the CouplingElement.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.couplingPortRole		
Mapping Rule		Mapping Type
hostPort maps to ETHSWT_HOST_PORT.		full
upLinkPort maps to ETHSWT_UP_LINK_PORT.		
standardPort maps to non configured EthSwtPortRole.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00101]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort	
BSW Parameter		BSW Type
EthSwtPortTimeStampSupport		EcucBooleanParamDef
BSW Description		
Enables/Disables the Switch-port specific timestamping.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00112]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort	
BSW Parameter		BSW Type
EthSwtPortTrcvRef		EcucReferenceDef
BSW Description		
Reference to the Ethernet transceiver driver this EthSwtPort is connected with.		
Template Description		
Specifies the physical layer type of the CouplingPort.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.physicalLayerType		

Mapping Rule		Mapping Type
<p>The reference shall be set if the CouplingPort for which the EthSwtPort is created has a defined physicalLayerType.</p> <p>The value of the parameter EthTrcvPhysLayerType that is defined within the referenced EthTrcvConfig container shall be derived from CouplingPort.physicalLayerType.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00041]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort	
BSW Parameter		BSW Type
EthSwtPortVlanMembership		EcucParamConfContainerDef
BSW Description		
Description Determines the membership of this port to the virtual network, i.e. frames with this VID can be received and transmitted via this port.		
Template Description		
Static logical channel or VLAN binding to a switch-port.		
The reference to an EthernetPhysicalChannel without a VLAN defined represents the handling of untagged frames.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00079]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership	
BSW Parameter		BSW Type
EthSwtPortVlanDefaultPriority		EcucIntegerParamDef
BSW Description		
Determines the standard output-priority outgoing messages will be tagged with.		
Template Description		
Standard output-priority outgoing Frames will be tagged with.		
Defines the priority that received frames are assigned together with the VLAN Id (default-Vlan). The values from 0 (best effort) to 7 (highest) are allowed.		
In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.defaultPriority		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00056]	

BSW Module	BSW Context

EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership	
BSW Parameter		BSW Type
EthSwtPortVlanForwardingType	EcucEnumerationParamDef	
BSW Description		
Defines how the message with a specific VLAN Id shall be handled.		
Template Description		
Attribute denotes whether a VLAN tagged ethernet frame will be # sent with its VLAN tag (sentTagged) # sent without a VLAN tag (sentUntagged) # will be dropped at this port (notSent or VLAN not member of this list)		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.sendActivity		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwt_00026]

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership/EthSwtPortVlanForwardingType	
BSW Parameter		BSW Type
ETHSWT_NOT_SENT		EcucEnumerationLiteralDef
BSW Description		
The message with the specific VLAN Id shall not be sent at this port.		
Template Description		
will not be sent		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetSwitchVlanEgressTaggingEnum.notSent		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership/EthSwtPortVlanForwardingType	
BSW Parameter		BSW Type
ETHSWT_SENT_TAGGED		EcucEnumerationLiteralDef
BSW Description		
The message with the specific VLAN Id shall be sent with its VLAN Id at this port.		
Template Description		
sent with its VLAN tag		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetSwitchVlanEgressTaggingEnum.sentTagged		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership/EthSwtPortVlanForwardingType	
BSW Parameter		BSW Type
ETHSWT_SENT_UNTAGGED		EcucEnumerationLiteralDef
BSW Description		
The message with the specific VLAN Id shall sent untagged.		
Template Description		
sent without a VLAN tag		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetSwitchVlanEgressTaggingEnum.sentUntagged		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortVlanMembership	
BSW Parameter		BSW Type
EthSwtPortVlanMembershipld		EcucIntegerParamDef
BSW Description		
Determines the VID of the virtual network this port belongs to.		
Template Description		
A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanConfig.vlanIdentifier		
Mapping Rule		Mapping Type
if a EthernetPhysicalChannel.vlan.vlanIdentifier is defined the value of vlanIdentifier shall be used for EthSwtPortVlanMembershipld. If no EthernetPhysicalChannel.vlan or EthernetPhysicalChannel.vlan.vlanIdentifier is defined then EthSwtPortVlanMembershipld shall be set to 0.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwt_00080]

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtConfig	
BSW Parameter		BSW Type
EthSwtSpi		EcucParamConfContainerDef
BSW Description		
Configuration of one Ethernet Switch SPI access (if SPI is used).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwt_00030]

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTSpi	
BSW Parameter		BSW Type
EthSwTSpiSequence		EcucParamConfContainerDef
BSW Description		
<p>Container gives EthSwT driver information about one SPI sequence.</p> <p>One SPI sequence used by EthSwT driver is in exclusive use for it. No other driver is allowed to access this sequence. EthSwT driver may use one sequence to access n EthSwT hardware chips of the same type or n sequences are used to access one single EthSwT hardware chip.</p> <p>If a EthSwT hardware has no SPI interface, there is no instance of this container.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwT_00034]

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTSpi/EthSwTSpiSequence	
BSW Parameter		BSW Type
EthSwTSpiAccessSynchronous		EcucBooleanParamDef
BSW Description		
<p>This parameter is used to define whether the access to the Spi sequence is synchronous or asynchronous.</p> <p>true: SPI access is synchronous. false: SPI access is asynchronous.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwT_00036]

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTConfig/EthSwTSpi/EthSwTSpiSequence	
BSW Parameter		BSW Type
EthSwTSpiSequenceName		EcucReferenceDef
BSW Description		
Reference to a Spi sequence configuration container.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthSwT_00035]

BSW Module		BSW Context	
EthSwT		EthSwT	
BSW Parameter		BSW Type	
EthSwTGeneral		EcucParamConfContainerDef	
BSW Description			
General configuration of Ethernet Switch Driver module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00003]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTDeletePortMirrorConfigurationApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_DeletePortMirrorConfiguration API			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00133]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00002]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	

EthSwtEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the Ethernet switch driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet switch driver will operate as an independent instance in each of the partitions.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00129]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtGeneral
BSW Parameter	BSW Type
EthSwtEnableCableDiagnosticApi	EcucBooleanParamDef
BSW Description	
Enable/disable the APIs for cable diagnostic: EthSwt_RunPortCableDiagnostic, EthSwt_GetPortCableDiagnosticsResult	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00135]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtGeneral
BSW Parameter	BSW Type
EthSwtEnableVlanApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwt_EnableVLAN API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00055]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtGeneral
BSW Parameter	BSW Type
EthSwtGetAriTableApi	EcucBooleanParamDef
BSW Description	

Enables / Disables EthSwT_GetArITable API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00052]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetBaudRateApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_GetBaudRate API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00121]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetCfgDataRawDone	EcucFunctionNameDef
BSW Description	
Defines the function name for <GetCfgDataRawDone>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00124]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetCfgRaw	EcucBooleanParamDef
BSW Description	
Disable /Enable support of reading raw data from switch memory	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00123]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type
EthSwTGetCounterValuesApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwT_GetCounterValues API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00053]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type
EthSwTGetDuplexModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwT_GetDuplexMode API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00122]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type
EthSwTGetLinkStateApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwT_GetLinkState API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00120]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetMacLearningModeApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_GetMacLearningMode API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00061]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetMaxFIFOBufferFillLevelApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_GetMaxFIFOBufferFillLevel API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00131]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetPortCableDiagnosticsResultApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_GetPortCableDiagnosticsResult API			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00092]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetPortIdentifierApi		EcucBooleanParamDef	
BSW Description			

Enables / Disables EthSwT_GetPortIdentifier API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00083]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetPortMacAddrApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_GetPortMacAddr API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00051]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetPortMirrorStateApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_GetPortMirrorState API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00087]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTGetPortSignalQualityApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_GetPortSignalQuality API	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00082]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter		BSW Type
EthSwtGetRxStatsApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwt_GetRxStats API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00065]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter		BSW Type
EthSwtGetSwitchIdentifierApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwt_GetSwitchIdentifier API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00084]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter		BSW Type
EthSwtGetSwitchPortModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthSwt_GetSwitchPortMode API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00118]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetSwitchRegApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_GetSwitchReg API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00066]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetTxErrorCounterValuesApi		EcucBooleanParamDef	
BSW Description			
Enables/Disables Eth_GetTxErrorCounterValues API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00100]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGetTxStatsApi		EcucBooleanParamDef	
BSW Description			
Enables/Disables Eth_GetTxStats API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00099]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTGlobalTimeSupportApi		EcucBooleanParamDef	
BSW Description			

Enables/Disables the Global Time APIs used amongst others by Global Time Synchronization over Ethernet.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00107]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTIndex	EcucIntegerParamDef
BSW Description	
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00033]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTLinkDownCallout	EcucFunctionNameDef
BSW Description	
Defines the function name for the <EthSwTLinkDownCallout> callout.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00115]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTLinkUpCallout	EcucFunctionNameDef
BSW Description	
Defines the function name for the <EthSwTLinkUpCallout> callout.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00116]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTLowPowerModeSupport	EcucBooleanParamDef
BSW Description	
Disable / Enable support of low power mode.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00102]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTMainFunctionPeriod	EcucFloatParamDef
BSW Description	
The cycle time of the periodic main function of EthSwT. Defined in seconds .	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00071]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTManagementSupportApi	EcucBooleanParamDef
BSW Description	
Enables/Disables the Switch management APIs to support a Switch-port specific communication attribute access.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00108]

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter	BSW Type	
EthSwTMgmtInfoIndicationTimeout	EcucFloatParamDef	
BSW Description		
<p>This parameter specifies the timeout while the Switch driver is waiting for management information out of the Switch for reception.</p> <p>The value 0 deactivates the timeout supervision.</p> <p>Unit: seconds</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00109]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter	BSW Type	
EthSwTPersistentConfigurationResult	EcucBooleanParamDef	
BSW Description		
<p>Enables / Disables the callback API <User>_PersistentConfigurationResult.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwT_00062]	

BSW Module	BSW Context	
EthSwT	EthSwT/EthSwTGeneral	
BSW Parameter	BSW Type	
EthSwTPersistentConfigurationResultCallback	EcucFunctionNameDef	
BSW Description		
<p>Defines the function name for <EthSwTPersistentConfigurationResultCallback>.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00063]

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter	BSW Type	
EthSwtPublicCddHeaderFile	EcucStringParamDef	
BSW Description		
Defines header files for callback functions which shall be included in case of CDDs.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00064]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter	BSW Type	
EthSwtReadPortMirrorConfigurationApi	EcucBooleanParamDef	
BSW Description		
Enables / Disables EthSwt_ReadPortMirrorConfiguration API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00086]	

BSW Module	BSW Context	
EthSwt	EthSwt/EthSwtGeneral	
BSW Parameter	BSW Type	
EthSwtReadTrcvRegisterApi	EcucBooleanParamDef	
BSW Description		
Enables / Disables EthSwt_ReadTrcvRegister API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthSwt_00069]	

BSW Module	BSW Context
------------	-------------

EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTResetConfigurationApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_ResetConfiguration API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00049]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetForwardingModeApi	EcucBooleanParamDef
BSW Description	
Enables /disables EthSwT_SetForwardingMode API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00104]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetMacLearningModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetMacLearningMode API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00060]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetPortLoopbackModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetPortLoopbackModeApi API	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00090]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetPortMirrorStateApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetPortMirrorState API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00088]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetPortTestModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetPortTestMode API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00089]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetPortTxModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetPortTxModeApi API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00091]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetSwitchPortModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetSwitchPortMode API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00117]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTSetSwitchRegApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_SetSwitchReg API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00067]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTStartSwitchPortAutoNegotiationApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_StartSwitchPortAutoNegotiation API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00119]

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTStoreConfigurationApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthSwT_StoreConfiguration API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00050]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTVerifyConfigApi		EcucBooleanParamDef	
BSW Description			
Enables /disables EthSwT_VerifyConfig API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00105]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables version info API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthSwT_00031]	

BSW Module		BSW Context	
EthSwT		EthSwT/EthSwTGeneral	
BSW Parameter		BSW Type	
EthSwTWritePortMirrorConfigurationApi		EcucBooleanParamDef	
BSW Description			

Enables / Disables EthSwT_WritePortMirrorConfiguration API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00085]

BSW Module	BSW Context
EthSwT	EthSwT/EthSwTGeneral
BSW Parameter	BSW Type
EthSwTWriteTrcvRegisterApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthSwT_WriteTrcvRegister API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwT_00070]

C.6.4 Service Discovery

BSW Module	BSW Context
Sd	Sd
BSW Parameter	BSW Type
SdConfig	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR Service Discovery module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00003]

BSW Module	BSW Context
Sd	Sd/SdConfig
BSW Parameter	BSW Type
SdCapabilityRecordMatchCallout	EcucParamConfContainerDef
BSW Description	

Callout that is invoked by the Sd implementation to determine whether the configuration options contained in the entries of a received SD message match the capability record elements configured in SdServerCapabilityRecord or SdClientCapabilityRecord.

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00124]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdCapabilityRecordMatchCallout
BSW Parameter	BSW Type
SdCapabilityRecordMatchCalloutName	EcucFunctionNameDef
BSW Description	
Function name (i.e., C-identifier) of the SdCapabilityRecordMatchCallout.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00125]

BSW Module	BSW Context
Sd	Sd/SdConfig
BSW Parameter	BSW Type
SdInstance	EcucParamConfContainerDef
BSW Description	
This container represents an instance of the SD; i.e. the SD configuration for a certain link.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00084]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	BSW Type
SdClientService	EcucParamConfContainerDef
BSW Description	
This container specifies all parameters used by Client services.	
Template Description	

Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance	
Mapping Rule	Mapping Type
Create container for each existing ConsumedServiceInstance that is available in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00005]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdBlacklistedVersions		EcucParamConfContainerDef
BSW Description		
Collection of blacklisted versions.		
Template Description		
Collection of blacklisted versions.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.blacklistedVersion		
Mapping Rule	Mapping Type	
If at least one ConsumedServiceInstance.blacklistedVersion exists the container shall be derived.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00141]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdBlacklistedVersions	
BSW Parameter		BSW Type
SdBlacklistedMinorVersions		EcucIntegerParamDef
BSW Description		
Blacklisted MinorVersions.		
Template Description		
Minor Version of the ServiceInterface.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipServiceVersion.minorVersion		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00142]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdClientCapabilityRecord		EcucParamConfContainerDef
BSW Description		

Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.	
The following use cases are supported:	
1) Key present, with no value (e.g. "passreq" -- password required for this service)	
2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed)	
3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")	
Template Description	
A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.capabilityRecord	
Mapping Rule	Mapping Type
1:1 mapping to ConsumedServiceInstance.capabilityRecord.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00072]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord
BSW Parameter	BSW Type
SdClientServiceCapabilityRecordKey	EcucStringParamDef
BSW Description	
Defines a CapabilityRecord key.	
Template Description	
Defines a key.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00073]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord
BSW Parameter	BSW Type
SdClientServiceCapabilityRecordValue	EcucStringParamDef
BSW Description	
Defines the corresponding CapabilityRecord value.	
Template Description	
Defines the corresponding value.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00074]

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type	
SdClientCapabilityRecordMatchCalloutRef		EcucReferenceDef	
BSW Description			
Reference to a SdCapabilityRecordMatchCallout, The referenced SdCapabilityRecordMatchCallout is invoked to determine whether the configuration options contained in the entries of a received SD message match the client's configured SdClientCapabilityRecord elements.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00127]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type	
SdClientServiceAutoRequire		EcucBooleanParamDef	
BSW Description			
If existing and set to true, this Service will be set to "required" on start.			
Template Description			
Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.autoRequire			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00143]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type	
SdClientServiceHandleId		EcucIntegerParamDef	
BSW Description			
The HandleId by which the BswM can identify this Client Service Instance.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00079]	

BSW Module	BSW Context
------------	-------------

Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdClientServiceId		EcucIntegerParamDef
BSW Description		
Id to identify the service. This is unique for the service interface.		
Template Description		
This attribute represents the ability to describe the SOME/IP service ID that is searched.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.service Identifier		
Mapping Rule		Mapping Type
Shall be derived from ConsumedServiceInstance.serviceIdentifier		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00020]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdClientServiceInstanceId		EcucIntegerParamDef
BSW Description		
Configuration parameter to specify Instance Id of the service as used in SD entries.		
Template Description		
This attribute represents the ability to describe the required service instance ID.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.instance Identifier		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00022]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdClientServiceMajorVersion		EcucIntegerParamDef
BSW Description		
Major version number of the Service as used in the SD entries.		
Template Description		
Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.majorVersion		
Mapping Rule		Mapping Type
Shall be derived from ConsumedServiceInstance.majorVersion		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00070]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type

SdClientServiceMinorVersion	EcucIntegerParamDef
BSW Description	
Minor version number of the Service as used in the SD Service Entries. If configured to 0xffffffff (any), SD will accept all Minor Versions.	
Template Description	
Minor Version of the ServiceInterface. Value can be set to a number that represents the Minor Version of the searched service or to ANY.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.minorVersion	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00071]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	BSW Type
SdClientServiceTcpRef	EcucReferenceDef
BSW Description	
Reference to the SoAdSocketConnection representing the data path (TCP) for communication with methods.	
This element is also used to set the remote address of the server and to open the TCP connection.	
Template Description	
The local address over which the CSI is consumed (udp, tcp or both).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress	
Mapping Rule	Mapping Type
This reference shall be derived from the ConsumedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port. If the ConsumedServiceInstance does not reference a TCP Port with the localUnicastAddress reference the SdServerServiceTcpRef shall be skipped.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00100]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	BSW Type
SdClientServiceTimerRef	EcucReferenceDef
BSW Description	
The reference of the SdClientTimer container for this service.	
Template Description	
Client specific configuration settings relevant for the SOME/IP service discovery.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.sdClientTimerConfig	
Mapping Rule	Mapping Type

The reference to the SdClientTimer shall be created pointing to the SdClient Timer container which was created based on the ConsumedServiceInstance.sd ClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00103]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter	BSW Type	
SdClientServiceUdpRef	EcucReferenceDef	
BSW Description		
Reference to the SoAdSocketConnection representing the data path (UDP) for communication with methods.		
This element is also used to set the remote address of the server.		
Template Description		
The local address over which the CSI is consumed (udp, tcp or both).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress		
Mapping Rule		Mapping Type
This reference shall be derived from the ConsumedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port. If the ConsumedServiceInstance does not reference a UDP Port with the localUnicastAddress reference the SdServerServiceTcpRef shall be skipped.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00101]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter	BSW Type	
SdConsumedEventGroup	EcucParamConfContainerDef	
BSW Description		
A Service may have event groups which can be consumed. A service consumer has to subscribe to the corresponding event-group. After the subscription the event consumer takes the role of a server and the event provider that of a client.		
Template Description		
This element represents an event-group to which the service consumer wants to subscribe.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup		
Mapping Rule		Mapping Type
Create container for every existing consumedEventGroup that is aggregated by the ConsumedServiceInstance		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00056]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter	BSW Type	
SdConsumedEventGroupAutoRequire	EcucBooleanParamDef	
BSW Description		

If existing and set to true, this EventGroup will be set to "required" on start.	
Template Description	
Defines that this ConsumedEventGroup shall be requested (subscribed) as soon as the corresponding ConsumedServiceInstance is requested. This could be at ECU start, if ConsumedServiceInstance.autoRequire is set to TRUE or as soon as the ConsumedServiceInstance is requested by the application, if ConsumedServiceInstance.autoRequire is set to FALSE.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.autoRequire	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00144]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupHandleId	EcucIntegerParamDef
BSW Description	
The HandleId by which the BswM can identify this EventGroup.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00116]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupId	EcucIntegerParamDef
BSW Description	
The Eventgroup Id of this eventGroup as a unique identifier of the eventgroup in this service. This identifier is used for EventGroup entries as well. Please note, that the Eventgroup ID 0x0000 is reserved.	
Template Description	
EventGroup ID. Shall be unique within one system to allow service discovery.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.eventGroup Identifier	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00057]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupMulticastActivationRef	EcucReferenceDef

BSW Description	
<p>The reference of a Routing Group in order to activate and setup the Socket Connection for Multicast Events of this EventGroup. The multicast address from the received Multicast option is setup by SoAd_RequestIpAddrAssignment.</p> <p>The local address is the same as for the unicast events; thus, it was sent in the UDP End-point option of the Subscribe EventGroup entry.</p> <p>This is usually equal to the SdConsumedEventGroupUdpActivationRef.</p>	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	Mapping Type
This container shall be created if the EventHandler aggregates a PduActivationRoutingGroup with eventGroupControlType = activationMulticast.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00106]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupMulticastGroupRef	EcucReferenceDef
BSW Description	
Reference to the SoAdSocketConnectionGroup representing the multicast data path (UDP).	
Template Description	
Client specific configuration settings relevant for the SOME/IP service discovery.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.sdClientTimerConfig	
Mapping Rule	Mapping Type
<p>ToDo: How to derive this parameter?</p> <p>New flag in the ConsumedServiceInstance: Name: numberOfSupportedMulticastAddresses Desc: "Number of Multicast Addresses that this CSI is able to deal with. This attribute is used in the Ecuc to reserve resources for the Multicast communication (Derivation of SdConsumedEventGroupMulticastGroupRef).</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00119]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupTcpActivationRef	EcucReferenceDef
BSW Description	

<p>The reference of the Routing Group for activation of the data path for receiving TCP events.</p> <p>This element is also being used for getting the IP address and port number for building the TCP endpoint option for the Subscribe EventGroup entry.</p> <p>If no TCP methods are used in the service, this element is also being used for setting the remote address (TCP Endpoint option referenced by the Offer Service entry) and opening the TCP connection to the server before sending the Subscribe EventGroup entry. If multiple EventGroups of the same Service Instance are subscribed the TCP connection will be shared and must be opened only once.</p>	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	Mapping Type
This container shall be created if the ConsumedEventGroup aggregates a Pdu ActiationRoutingGroup with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPdulIdentifierTcp reference to a collection of SoConIPdulIdentifiers.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00105]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupTimerRef	EcucReferenceDef
BSW Description	
The reference of the SdClientTimer container for this eventGroup.	
Template Description	
Client Timing configuration settings that are EventGroup specific.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.sdClientTimerConfig	
Mapping Rule	Mapping Type
The reference to the SdServerTimer shall be created pointing to the SdServerTimer container which was created based on the ConsumedEventGroup.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00107]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	BSW Type
SdConsumedEventGroupUdpActivationRef	EcucReferenceDef
BSW Description	

The reference of the Routing Group for activation of the data path for receiving UDP events.	
This element is also being used for getting the IP address and port number for building the UDP endpoint option for the Subscribe EventGroup entry.	
If no UDP methods are used in the service, this element is also being used for setting the remote address (UDP Endpoint option referenced by the Offer Service entry). If multiple EventGroups of the same Service Instance are subscribed the UDP Socket Connection will be shared and must be set only once.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	Mapping Type
This container shall be created if the ConsumedEventGroup aggregates a PduActiationRoutingGroup with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPdulIdentifierUdp reference to a collection of SoConIPdulIdentifiers.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00104]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	BSW Type
SdConsumedMethods	EcucParamConfContainerDef
BSW Description	
Container element for representing the data path for accessing the server methods.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	Mapping Type
If the ConsumedServiceInstance contains a methodActivationRoutingGroup then this container shall be created	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00099]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedMethods
BSW Parameter	BSW Type
SdClientServiceActivationRef	EcucReferenceDef
BSW Description	
Reference to a SoAdRoutingGroupRef to activate/deactivate the data path for the methods.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	Mapping Type
If the ConsumedServiceInstance contains a methodActivationRoutingGroup then this reference shall be created	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00102]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdServiceGroupRef		EcucReferenceDef
BSW Description		
Reference to the SdServiceGroupS this SdClientService belongs to.		
Template Description		
This reference assigns a set of ProvidedServiceInstances to the ConsumedProvidedServiceInstanceGroup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedProvidedServiceInstanceGroup.consumedServiceInstance		
Mapping Rule		Mapping Type
Find ConsumedProvidedServiceInstanceGroup that points to this ConsumedServiceInstance and create the reference.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00137]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		BSW Type
SdVersionDrivenFindBehavior		EcucEnumerationParamDef
BSW Description		
Defined the possible acceptance kinds for required service instances.		
Template Description		
Defines the service discovery find behavior.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.versionDrivenFindBehavior		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00140]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		BSW Type
SdClientTimer		EcucParamConfContainerDef
BSW Description		
This container specifies all timers used by the Service Discovery module for Client Services.		
Template Description		
Client specific configuration settings relevant for the SOME/IP service discovery.		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.sdClientTimerConfig	
Mapping Rule	Mapping Type
The Timing parameters can be derived from the SomeipSdClientServiceInstanceConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig. If multiple ConsumedServiceInstances reference the same SomeipSdClientServiceInstanceConfig, then one SdClientTimer shall be derived for those ConsumedServiceInstances which are referenced by the same ConsumedProvidedServiceInstanceGroup and for those which have ConsumedServiceInstance.autoRequire set to true. For all other ConsumedServiceInstances an own SdClientTimer shall be derived and referenced by the corresponding SdServerService.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00043]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerInitialFindDelayMax	EcucFloatParamDef
BSW Description	
Max value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
Template Description	
<p>@RESTRICT_TO_STANDARD:CP! Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! Max Value in seconds to delay randomly the first offer (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the transmission of a find message (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialDelayMax Value	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00063]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerInitialFindDelayMin	EcucFloatParamDef
BSW Description	
Min value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
Template Description	

@RESTRICT_TO_STANDARD:CP! Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!	
@RESTRICT_TO_STANDARD:AP! Min Value in seconds to delay randomly the first offer (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the transmission of a find message (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialDelayMin Value	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00044]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerInitialFindRepetitionsBaseDelay	EcucFloatParamDef
BSW Description	
The base delay in [s] for find repetitions. Successive finds have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ClientService.	
Template Description	
@RESTRICT_TO_STANDARD:CP! The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay. @END_RESTRICT_TO_STANDARD!	
@RESTRICT_TO_STANDARD:AP! The base delay for offer repetitions (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or find repetitions (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). Successive find messages have an exponential back off delay. @END_RESTRICT_TO_STANDARD!	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialRepetitions BaseDelay	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00047]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerInitialFindRepetitionsMax	EcucIntegerParamDef
BSW Description	
Configuration for the maximum number of find repetitions. This parameter is mandatory for ClientService.	

Template Description	
<p>@RESTRICT_TO_STANDARD:CP! Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! Describes the maximum amount of offer repetitions (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the maximum amount of find repetitions (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialRepetitionsMax	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00046]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerRequestResponseMaxDelay	EcucFloatParamDef
BSW Description	
Maximum allowable response delay to entries received by multicast in seconds. This parameter is mandatory for ConsumedEventGroups.	
Template Description	
Maximum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::RequestResponseDelay.maxValue	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00036]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	BSW Type
SdClientTimerRequestResponseMinDelay	EcucFloatParamDef
BSW Description	
Minimum allowable response delay to the find message in seconds. This parameter is mandatory for ConsumedEventGroups.	
Template Description	
Minimum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::RequestResponseDelay.minValue	
Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00064]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter	BSW Type	
SdClientTimerTTL	EcucIntegerParamDef	
BSW Description		
Time to live for find and subscribe messages.		
Template Description		
This attribute represents the ability to define the time in seconds the service find is valid.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipSdClientServiceInstanceConfig.serviceFindTimeToLive		
Mapping Rule		Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00075]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter	BSW Type	
SdSubscribeEventgroupRetryDelay	EcucFloatParamDef	
BSW Description		
Time in seconds when a subscription to an event group shall be retrigged, if no SubscribeEventGroupAck or SubscribeEventGroupNack was received.		
Template Description		
This attribute defines the interval in seconds to re-trigger a subscription to a Eventgroup, if a retry to subscribe to a Eventgroup is configured (subscribeEventgroupRetryMax > 0).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipSdClientEventGroupTimingConfig.subscribeEventgroupRetryDelay		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00133]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter	BSW Type	
SdSubscribeEventgroupRetryMax	EcucIntegerParamDef	
BSW Description		
Maximum count of retry a subscription, if a subscription to an event group is not acknowledged by SubscribeEventGroupAck or SubscribeEventGroupNack. 0x0=no retry, 0xFF=retry forever (as long as the event group is requested)		
Template Description		
This attribute define the maximum counts of retries to subscribe to an Eventgroup. If the value is set to 0 no retry shall be done. If the value is set to 255 the retry shall be done as along as the Eventgroup is requested and no SubscribeEventGroupAck was received.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipSdClientEventGroupTimingConfig.subscribeEventgroupRetryMax		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00132]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter	BSW Type	
SdInstanceDemEventParameterRefs	EcucParamConfContainerDef	
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00120]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdInstanceDemEventParameterRefs	
BSW Parameter	BSW Type	
SD_E_MALFORMED_MSG	EcucReferenceDef	
BSW Description		
Reference to the DemEventParameter which shall be issued when the SD Instance received malformed message.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00121]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdInstanceDemEventParameterRefs	
BSW Parameter	BSW Type	
SD_E_OUT_OF_RES	EcucReferenceDef	
BSW Description		
Reference to the DemEventParameter which shall be issued when the SD Instance does not have enough resources to handle client.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00122]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdInstanceDemEventParameterRefs	
BSW Parameter	BSW Type	
SD_E_SUBSCR_NACK_RECV	EcucReferenceDef	
BSW Description	Reference to the DemEventParameter which shall be issued when receiving SubscribeEventgroup-Nack entry.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00123]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter	BSW Type	
SdInstanceHostname	EcucStringParamDef	
BSW Description	Configuration parameter to specify the Hostname.	
Template Description	Defines the fully qualified domain name (FQDN) e.g. some.example.host.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpoint.fullyQualifiedDomain Name	
Mapping Rule	Mapping Type	
Shall be derived from the NetworkEndpoint.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00012]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter	BSW Type	
SdInstanceLocalAdressCheckLength	EcucIntegerParamDef	
BSW Description	<p>This item describes on how many bits of the addresses shall be compared to determine, if a remote address is acceptable to be used.</p> <p>This shall support IPv4 (0..32) and IPv6 (0..128). If this item is not present, the security checks use the configured netmask instead.</p> <p>"0" meaning not to check at all. For example "8" means that the first 8 bits of a remote address must be equal to the local address to be considered acceptable.</p>	
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00128]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		BSW Type
SdInstanceMulticastRxPdu		EcucParamConfContainerDef
BSW Description		
This container specifies the received PDU.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort		
Mapping Rule		Mapping Type
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00081]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdInstanceMulticastRxPdu	
BSW Parameter		BSW Type
SdRxPduId		EcucIntegerParamDef
BSW Description		
ID of the PDU that will be received via the API Sd_SoAdIfRxIndication().		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00028]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdInstanceMulticastRxPdu	
BSW Parameter		BSW Type
SdRxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00029]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance	
BSW Parameter		BSW Type	
SdInstanceTxPdu		EcucParamConfContainerDef	
BSW Description			
This container specifies the transmitted PDU.			
Template Description			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort			
Mapping Rule			Mapping Type
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00030]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdInstanceTxPdu	
BSW Parameter		BSW Type	
SdTxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00109]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance	
BSW Parameter		BSW Type	
SdInstanceUnicastRxPdu		EcucParamConfContainerDef	
BSW Description			
This container specifies the received PDU.			
Template Description			
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort			
Mapping Rule			Mapping Type
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00027]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdInstanceUnicastRxPdu	
BSW Parameter		BSW Type	

SdRxPduId	EcucIntegerParamDef
BSW Description	
ID of the PDU that will be received via the API Sd_SoAdIfRxIndication().	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_SD_00082]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdInstanceUnicastRxPdu
BSW Parameter	BSW Type
SdRxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_SD_00083]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	BSW Type
SdServerService	EcucParamConfContainerDef
BSW Description	
This container specifies all parameters used by Server services.	
Template Description	
Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance	
Mapping Rule	
Mapping Type	
Create container for each existing ProvidedServiceInstance that is available in the Ecu Extract.	full
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_SD_00004]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdEventHandler	EcucParamConfContainerDef
BSW Description	
Container Element for representing an EventGroup as part of the Service Instance.	

Template Description	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.eventHandler	
Mapping Rule	Mapping Type
Create container for every existing EventHandler that is aggregated by the ProvidedServiceInstance	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00055]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	BSW Type
SdEventHandlerEventGroupId	EcucIntegerParamDef
BSW Description	
The EventGroup Id of this EventGroup as a unique identifier of the EventGroup in this service. This identifier is used for EventGroup entries as well. Please note, that the Eventgroup ID 0x0000 is reserved.	
Template Description	
Unique Identifier that identifies the EventGroup in SOME/IP. This Identifier is sent as Eventgroup ID in SOME/IP Service Discovery messages.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.eventGroupIdentifier	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00061]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	BSW Type
SdEventHandlerHandleId	EcucIntegerParamDef
BSW Description	
The HandleId by which the BswM can identify this EventGroup.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00112]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	BSW Type
SdEventHandlerMulticast	EcucParamConfContainerDef
BSW Description	

The subcontainer including the Routing Group for Activation of Events sent over Multicast.	
The activation ref is also being used for identification of the related Socket Connection in order to find the Multicast Address used in the Multicast Option referenced by the Subscribe EventGroup Ack entry.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.pduActivationRouting Group	
Mapping Rule	Mapping Type
This container shall be created if the EventHandler aggregates a PduActivation RoutingGroup with eventGroupControlType = activationMulticast.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00094]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandler Multicast
BSW Parameter	BSW Type
SdEventActivationRef	EcucReferenceDef
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.event GroupControlType	
Mapping Rule	Mapping Type
Create this reference in the SdEventHandlerUdp or SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00096]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandler Multicast
BSW Parameter	BSW Type
SdMulticastEventSoConRef	EcucReferenceDef

BSW Description	
Reference to the SoAdSocketConnection representing the multicast data path (UDP).	
Template Description	
Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the mulicastThreshold is exceeded.	
This address is transmitted in the SD-SubscribeEventGroupAck Message to client (answer to SD-SubscribeEventGroup).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.eventMulticastAddress	
Mapping Rule	Mapping Type
Create this reference if the EventHandler in the System Description contains an eventMulticastAddress reference to an ApplicationEndpoint that points to a NetworkEndpoint that in turn defines an IP MulticastAddress.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00118]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	BSW Type
SdEventHandlerMulticastThreshold	EcucIntegerParamDef
BSW Description	
Specifies the number of subscribed clients that trigger the Server to change the transmission of events to Multicast.	
If configured to 0 only unicast will be used.	
If configured to 1 the first client will be already served by multicast.	
If configured to 2 the first client will be served with unicast and as soon as the second client arrives both will be served by multicast.	
This does not influence the handling of initial events, which are served using unicast only.	
Template Description	
Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast.	
If configured to 0 only unicast will be used.	
If configured to 1 the first client will be already served by multicast.	
If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast.	
This does not influence the handling of initial events, which are served using unicast only.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.multicastThreshold	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00097]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	BSW Type

SdEventHandlerTcp	EcucParamConfContainerDef
BSW Description	
The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over TCP.	
The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to find the related client by iterating the SdEventHandlerTcp elements (remote address statically configured or automatically set by opening TCP connection before subscription).	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.pduActivationRoutingGroup	
Mapping Rule	
This container shall be created if the EventHandler aggregates a PduActiation RoutingGroup with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPdulIdentifierTcp reference to a collection of SoConIPdulIdentifiers.	Mapping Type
	full
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_SD_00093]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp
BSW Parameter	BSW Type
SdEventActivationRef	EcucReferenceDef
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	
Create this reference in the SdEventHandlerUdp or SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast.	Mapping Type
Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast.	full
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_SD_00096]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp	
BSW Parameter		BSW Type
SdEventTriggeringRef		EcucReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.		
Template Description		
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.eventGroupControlType		
Mapping Rule		Mapping Type
Create this reference if eventGroupControlType in the PduActivationRoutingGroup is set to triggerUnicast or activationAndTriggerUnicast.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00095]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		BSW Type
SdEventHandlerTimerRef		EcucReferenceDef
BSW Description		
The reference of the SdServerTimer container for this EventGroup.		
Template Description		
Server Timing configuration settings that are EventGroup specific.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.sdServerEgTimingConfig		
Mapping Rule		Mapping Type
The reference to the SdServerTimer shall be created pointing to the SdServerTimer container which was created based on the EventHandler.sdServerEgTimingConfig. If multiple ProvidedServiceInstances reference the same SomeipSdServerServiceInstanceConfig, then one SdServerTimer shall be derived for those ProvidedServiceInstances which are referenced by the same ConsumedProvidedServiceInstanceGroup and for those which have ProvidedServiceInstance.autoAvailable set to true. For all other ProvidedServiceInstances an own SdServerTimer shall be derived and referenced by the corresponding SdServerService.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00113]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		BSW Type
SdEventHandlerUdp		EcucParamConfContainerDef
BSW Description		

<p>The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over UDP.</p> <p>The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to set the remote address of the client or find the related client by iterating the SdEventHandlerUdp elements (remote address statically configured or automatically set by method call before subscription).</p>	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::EventHandler.pduActivationRoutingGroup	
Mapping Rule	Mapping Type
This container shall be created if the EventHandler aggregates a PduActivationRoutingGroup with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTriggerUnicast and this PduActivationRoutingGroup points with the iPduIdentifierUdp reference to a collection of SoConIPduIdentifiers.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00092]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp
BSW Parameter	BSW Type
SdEventActivationRef	EcucReferenceDef
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	
<p>This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed.</p> <p>Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	Mapping Type
<p>Create this reference in the SdEventHandlerUdp or SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast.</p> <p>Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00096]

BSW Module	BSW Context
-------------------	--------------------

Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp	
BSW Parameter		BSW Type
SdEventTriggeringRef		EcucReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.		
Template Description		
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.eventGroupControlType		
Mapping Rule		Mapping Type
Create this reference if eventGroupControlType in the PduActivationRoutingGroup is set to triggerUnicast or activationAndTriggerUnicast.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00095]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdProvidedMethods		EcucParamConfContainerDef
BSW Description		
Container element for representing the needed elements of the data path for the methods provided by the service.		
Template Description		
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.methodActivationRoutingGroup		
Mapping Rule		Mapping Type
If the ProvidedServiceInstance contains a methodActivationRoutingGroup then this container shall be created		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00087]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdProvidedMethods	
BSW Parameter		BSW Type
SdServerServiceActivationRef		EcucReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup to activated and deactivate the data path for methods of the service.		
Template Description		
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	Mapping Type
If the ProvidedServiceInstance contains a methodActivationRoutingGroup then this reference shall be created	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00090]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerCapabilityRecord	EcucParamConfContainerDef
BSW Description	
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.	
The following use cases are supported:	
1) Key present, with no value (e.g. "passreq" -- password required for this service)	
2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed)	
3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")	
Template Description	
A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.capabilityRecord	
Mapping Rule	Mapping Type
1:1 mapping to ProvidedServiceInstance.capabilityRecord.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00032]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord
BSW Parameter	BSW Type
SdServerCapabilityRecordKey	EcucStringParamDef
BSW Description	
Defines a CapabilityRecord key.	
Template Description	
Defines a key.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00033]

BSW Module	BSW Context
-------------------	--------------------

Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord	
BSW Parameter		BSW Type
SdServerCapabilityRecordValue		EcucStringParamDef
BSW Description		
Defines the corresponding CapabilityRecord value.		
Template Description		
Defines the corresponding value.		
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00034]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdServerCapabilityRecordMatchCalloutRef		EcucReferenceDef
BSW Description		
Reference to a SdCapabilityRecordMatchCallout, The referenced SdCapabilityRecordMatchCallout is invoked to determine whether the configuration options contained in the entries of a received SD message match the server's configured SdServerCapabilityRecord elements.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00126]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdServerServiceAutoAvailable		EcucBooleanParamDef
BSW Description		
If existing and set to true, this Service will be set to "Available" on start.		
Template Description		
Defines that this ProvidedServiceInstance shall be offered by the service discovery at ECU start.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.autoAvailable		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00138]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdServerServiceHandleId		EcucIntegerParamDef

BSW Description	
The Handleld by which the BswM can identify this Server Service Instance.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00110]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceId	EcucIntegerParamDef
BSW Description	
Id to identify the service. This is unique for the service interface.	
Template Description	
This attribute represents the ability to describe the SOME/IP service ID that is offered.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.serviceIdentifier	
Mapping Rule	Mapping Type
Shall be derived from ProvidedServiceInstance.serviceIdentifier	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00009]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceInstanceId	EcucIntegerParamDef
BSW Description	
Configuration parameter to specify Instance Id of the Service implemented by the Server Service.	
Template Description	
Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.instanceIdentifier	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00011]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceLoadBalancingPriority	EcucIntegerParamDef
BSW Description	
Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.	

Template Description	
Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.loadBalancingPriority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00129]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceLoadBalancingWeight	EcucIntegerParamDef
BSW Description	
Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.	
Template Description	
Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.loadBalancingWeight	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00130]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceMajorVersion	EcucIntegerParamDef
BSW Description	
Major version number of the Service as used in SD Entries.	
Template Description	
Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::AbstractServiceInstance.majorVersion	
Mapping Rule	Mapping Type
Shall be derived from ProvidedServiceInstance.majorVersion	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00068]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceMinorVersion	EcucIntegerParamDef
BSW Description	

Minor version number of the Service as used e.g. in Offer Service entries.	
Template Description	
Minor Version of the Service that is provided by this ProvidedServiceInstance.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.minorVersion	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00069]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceTcpRef	EcucReferenceDef
BSW Description	
Reference to SoAdSocketConnectionGroup used for methods.	
This is used to access the local IP address and port for building the endpoint option for offers of this service.	
Template Description	
The local address over which the CSI is consumed (udp, tcp or both).	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress	
Mapping Rule	Mapping Type
This reference shall be derived from the ProvidedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port. If the ProvidedServiceInstance does not reference a TCP Port with the localUnicastAddress reference the SdServerServiceTcpRef shall be skipped.	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00088]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	BSW Type
SdServerServiceTimerRef	EcucReferenceDef
BSW Description	
The reference of the SdServerTimer container for this service.	
Template Description	
Server specific configuration settings relevant for the SOME/IP service discovery.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.sdServerTimerConfig	
Mapping Rule	Mapping Type
The reference to the SdServerTimer shall be created pointing to the SdServerTimer container which was created based on the ProvidedServiceInstance.sdServerTimerConfig.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00086]

BSW Module	BSW Context
-------------------	--------------------

Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdServerServiceUdpRef		EcucReferenceDef
BSW Description		
Reference to SoAdSocketConnectionGroup used for methods.		
This is used to access the local IP address and port for building the endpoint option for offers of this service.		
Template Description		
The local address over which the CSI is consumed (udp, tcp or both).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress		
Mapping Rule		Mapping Type
This reference shall be derived from the ProvidedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port. If the ProvidedServiceInstance does not reference a UDP Port with the localUnicastAddress reference the SdServerServiceTcpRef shall be skipped.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00089]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		BSW Type
SdServiceGroupRef		EcucReferenceDef
BSW Description		
Reference to the SdServiceGroupS this SdServerService belongs to.		
Template Description		
This reference assigns a set of ConsumedServiceInstances to the ConsumedProvidedServiceInstanceGroup.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedProvidedServiceInstanceGroup.providedServiceInstance		
Mapping Rule		Mapping Type
Find ConsumedProvidedServiceInstanceGroup that points to this ProvidedServiceInstance and create the reference.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00136]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		BSW Type
SdServerTimer		EcucParamConfContainerDef
BSW Description		
This container specifies all timers used by the Service Discovery module for Server Services.		
Template Description		
Server specific configuration settings relevant for the SOME/IP service discovery.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.sdServerTimerConfig		
Mapping Rule		Mapping Type

The Timing parameters can be derived from the SomeipSdServerServiceInstanceConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00035]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter	BSW Type	
SdServerTimerInitialOfferDelayMax	EcucFloatParamDef	
BSW Description		
Max value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.		
Template Description		
<p>@RESTRICT_TO_STANDARD:CP! Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! Max Value in seconds to delay randomly the first offer (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the transmission of a find message (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialDelayMax Value		
Mapping Rule	Mapping Type	
Take information from ProvidedServiceInstance.sdServerTimerConfig	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00039]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter	BSW Type	
SdServerTimerInitialOfferDelayMin	EcucFloatParamDef	
BSW Description		
Min value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.		
Template Description		
<p>@RESTRICT_TO_STANDARD:CP! Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!</p> <p>@RESTRICT_TO_STANDARD:AP! Min Value in seconds to delay randomly the first offer (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the transmission of a find message (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialDelayMin Value		
Mapping Rule	Mapping Type	

Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00038]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter	BSW Type	
SdServerTimerInitialOfferRepetitionBaseDelay	EcucFloatParamDef	
BSW Description		
The base delay in [s] for offer repetitions. Successive offers have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ServerService.		
Template Description		
@RESTRICT_TO_STANDARD:CP! The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay. @END_RESTRICT_TO_STANDARD!		
@RESTRICT_TO_STANDARD:AP! The base delay for offer repetitions (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or find repetitions (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). Successive find messages have an exponential back off delay. @END_RESTRICT_TO_STANDARD!		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialRepetitionsBaseDelay		
Mapping Rule		Mapping Type
Take information from ProvidedServiceInstance.sdServerTimerConfig		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00041]	

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter	BSW Type	
SdServerTimerInitialOfferRepetitionsMax	EcucIntegerParamDef	
BSW Description		
Configure the maximum amount of offer repetition. This parameter is mandatory for ServerService.		
Template Description		
@RESTRICT_TO_STANDARD:CP! Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig). @END_RESTRICT_TO_STANDARD!		
@RESTRICT_TO_STANDARD:AP! Describes the maximum amount of offer repetitions (if aggregated in role initialOfferBehavior by SomeipSdServerServiceInstanceConfig) or the maximum amount of find repetitions (if aggregated in role initialFindBehavior by SomeipSdClientServiceInstanceConfig). @END_RESTRICT_TO_STANDARD!		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::InitialSdDelayConfig.initialRepetitionsMax		
Mapping Rule		Mapping Type

Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00040]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter		BSW Type
SdServerTimerOfferCyclicDelay		EcucFloatParamDef
BSW Description		
Interval between cyclic offers in the main phase. This parameter is mandatory for ServerService.		
Template Description		
Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipSdServerServiceInstanceConfig.offerCyclicDelay		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00076]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter		BSW Type
SdServerTimerRequestResponseMaxDelay		EcucFloatParamDef
BSW Description		
Maximum allowable response delay to entries received by multicast in seconds.		
Template Description		
Maximum allowable response delay to entries received by multicast in seconds.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::RequestResponseDelay.maxValue		
Mapping Rule		Mapping Type
Take information from ProvidedServiceInstance.requestResponseDelay		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00114]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter		BSW Type
SdServerTimerRequestResponseMinDelay		EcucFloatParamDef
BSW Description		
Minimum allowable response delay to entries received by multicast in seconds.		
Template Description		
Minimum allowable response delay to entries received by multicast in seconds.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::RequestResponseDelay.minValue		
Mapping Rule		Mapping Type
Take information from ProvidedServiceInstance.requestResponseDelay		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SD_00115]

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter		BSW Type	
SdServerTimerTTL		EcucIntegerParamDef	
BSW Description			
Time to live for offer service.			
Template Description			
Defines the time in seconds the service offer is valid.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SomeipSdServerServiceInstanceConfig.serviceOfferTimeToLive			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00037]	

BSW Module		BSW Context	
Sd		Sd/SdConfig	
BSW Parameter		BSW Type	
SdServiceGroup		EcucParamConfContainerDef	
BSW Description			
Contains the configuration parameters of the AUTOSAR SD module's SdServiceGroupS.			
Template Description			
The AUTOSAR ServiceDiscovery is able to start and to stop ClientServices and ServerServices, respectively, at runtime. A SdServiceGroup contains several ClientServices and ServerServices, respectively.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedProvidedServiceInstanceGroup			
Mapping Rule		Mapping Type	
Create container for each CoreCommunication::ConsumedProvidedServiceInstanceGroup that is contained in the ECU Extract.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00134]	

BSW Module		BSW Context	
Sd		Sd/SdConfig/SdServiceGroup	
BSW Parameter		BSW Type	
SdServiceGroupHandleId		EcucIntegerParamDef	
BSW Description			
The numerical value used as the ID of this SdServiceGroup. The SdServiceHandleId is required by the API calls to start and stop SdServiceGroupS.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SD_00135]	

BSW Module	BSW Context	
Sd	Sd	
BSW Parameter	BSW Type	
SdGeneral	EcucParamConfContainerDef	
BSW Description	This container lists the general configuration parameters for the Service Discovery module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00002]	

BSW Module	BSW Context	
Sd	Sd/SdGeneral	
BSW Parameter	BSW Type	
SdDevErrorDetect	EcucBooleanParamDef	
BSW Description	Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00006]	

BSW Module	BSW Context	
Sd	Sd/SdGeneral	
BSW Parameter	BSW Type	
SdMainFunctionCycleTime	EcucFloatParamDef	
BSW Description	This parameter defines the cycle time in seconds of the periodic calling of Sd main function.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SD_00008]	

BSW Module	BSW Context	
Sd	Sd/SdGeneral	
BSW Parameter	BSW Type	

SdSetRemAddrOfClientRxMulticastSoCon	EcucBooleanParamDef
BSW Description	
<p>If SdSetRemAddrOfClientRxMulticastSoCon is set to TRUE, the Service Discovery module shall choose an multicast socket connection which match to the received Endpoint option of the corresponding OfferService. If no particular socket connection exist, then an unused socket connection with its remote address set to wildcard shall be used and the remote address shall be updated accordingly. If SdSetRemAddrOfClientRxMulticastSoCon is set to FALSE, the Service Discovery shall choose an unused socket connection with its remote address set to wildcard and skip to update the remote address, i.e. the wildcard for the remote address is kept.</p> <p>Note: setting SdSetRemAddrOfClientRxMulticastSoCon to FALSE supports the re-use of a multicast socket connection for multiple ClientServices which are located on the same ECU and subscribed to ServerServices which are located on differen ECUs. The configuration of the ECU where the ClientServices are located, could be simplified by only configuring one socket connection within the multicast socket connection group.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00139]

BSW Module	BSW Context
Sd	Sd/SdGeneral
BSW Parameter	BSW Type
SdSubscribeEventgroupRetryEnable	EcucBooleanParamDef
BSW Description	
<p>Switch to enable or disable the retry functionality to subscribe to Eventgroups of ServerServices with TTL set to 0xFFFFFFFF.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00131]

BSW Module	BSW Context
Sd	Sd/SdGeneral
BSW Parameter	BSW Type
SdVersionInfoApi	EcucBooleanParamDef
BSW Description	
<p>Enables and disables the version info API.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_SD_00007]

C.6.5 SoAd

BSW Module	BSW Context	
SoAd	SoAd	
BSW Parameter	BSW Type	
SoAdBswModules	EcucParamConfContainerDef	
BSW Description	Each container describes a specific BSW module that the SoAd shall interface to.	
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00102]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter	BSW Type	
SoAdBswModuleRef	EcucForeignReferenceDef	
BSW Description	<p>This is a reference to one BSW module's configuration (i.e. not the ECUC parameter definition template).</p> <p>Example, there could be several configurations of PduR and this reference selects one of them.</p> <p>SoAd has to figure out from the structure of the referenced BSW module's configuration, what kind of upper layer he deals with.</p> <p>In case of a CDD SoAd expects UL-APIs in form of <code>_SoAd<lf Tp><function></code> and expects CDD Pdu configuration structures according to the Ecu Configuration specification (chapter CDD module\Socket Adaptor).</p> <p>In case it is one of the standardized AUTOSAR BSW modules, the configuration structures and API names for interaction with SoAd are defined in the corresponding SWS.</p>	
Template Description	<p>Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure.</p> <p>As part of the BSW module description, the <code>EcucModuleConfigurationValues</code> element has two different roles:</p> <p>The <code>recommendedConfiguration</code> contains parameter values recommended by the BSW module vendor.</p> <p>The <code>preconfiguredConfiguration</code> contains values for those parameters which are fixed by the implementation and cannot be changed.</p> <p>These two <code>EcucModuleConfigurationValues</code> are used when the base <code>EcucModuleConfigurationValues</code> (as part of the base ECU configuration) is created to fill parameters with initial values.</p>	
M2 Parameter		

ECUCDescriptionTemplate::EcucModuleConfigurationValues	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00124]

BSW Module	BSW Context
SoAd	SoAd/SoAdBswModules
BSW Parameter	BSW Type
SoAdIf	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the Communication Interface APIs or not. Value true means that the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00104]

BSW Module	BSW Context
SoAd	SoAd/SoAdBswModules
BSW Parameter	BSW Type
SoAdIfTriggerTransmit	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the TriggerTransmit API or not. Value true means that the API is supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00145]

BSW Module	BSW Context
SoAd	SoAd/SoAdBswModules
BSW Parameter	BSW Type
SoAdIfTxConfirmation	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the TxConfirmation API or not. Value true means that the API is supported.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00106]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		BSW Type
SoAdLocalIpAddrAssignmentChg		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the LocalIpAddrAssignmentChg API or not. Value true means that the API is supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00143]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		BSW Type
SoAdSoConModeChg		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the SoConModeChg API or not. Value true means that the API is supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00107]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		BSW Type
SoAdTp		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TransportProtocol APIs or not. Value true means that the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00105]

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		BSW Type
SoAdUseCallerInfix		EcucBooleanParamDef
BSW Description		
Specifies if SoAd shall use (TRUE) the infix "SoAd" when calling an upper layer module function or not (FALSE). E.g. if SoAdUseCallerInfix is TRUE for the upper layer "ABC" then SoAd will call ABC_SoAdIfRxIndication() otherwise SoAd would call ABC_IfRxIndication().		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00128]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		BSW Type
SoAdUseTypeInfix		EcucBooleanParamDef
BSW Description		
Specifies if SoAd shall use (TRUE) the API type infix "Tp" or "If" when calling an upper layer module function or not (FALSE). E.g. if SoAdUseTypeInfix is TRUE for the upper layer "ABC" then SoAd will call ABC_IfRxIndication(), otherwise SoAd would call ABC_RxIndication().		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00129]	

BSW Module	BSW Context	
SoAd	SoAd	
BSW Parameter		BSW Type
SoAdConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR SoAd module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00103]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig
BSW Parameter	BSW Type
SoAdPduRoute	EcucParamConfContainerDef
BSW Description	
Describes the path of a PDU from an upper layer of the SoAd to the socket in the TCP/IP stack for transmission. This PDU can consume meta data items of type SOCKET_CONNECTION_ID_16.	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduTriggering	
Mapping Rule	Mapping Type
The SoAdPduRoute container shall be created for every PduTriggering that is transmitted by the regarded ECU. The information whether the Pdu is received or transmitted over a SocketConnection shall be derived from the PduTriggering element. The PduTriggering element contains references to IPduPorts of an EcuInstance. The IPduPort element contains a communicationDirection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00007]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	BSW Type
SoAdPduRouteDest	EcucParamConfContainerDef
BSW Description	
Specifies the PDU route destination.	
Template Description	
ApplicationEndpoint: An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.	
StaticSocketConnection.iPdulIdentifier: Assignment of IPdulIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier	
Mapping Rule	Mapping Type

<p>SOME/IP communication: If the regarded EcuInstance that transmits the PduTriggering is referenced by an ApplicationEndpoint that in turn is referenced by a ProvidedServiceInstance that has a relationship to the PduTriggering via a PduActivationRoutingGroup and this ApplicationEndpoint contains a maxNumberOfConnections setting then this container shall be created as many times as the maxNumberOfConnections value indicates.</p> <p>StaticSocketConnection: If the PduTriggering that is transmitted by the Ecu has a relationship to a StaticSocketConnection via a SoConIPdulIdentifier then this container shall be created once for this PduTriggering.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00119]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		BSW Type
SoAdTxPduHeaderId		EcucIntegerParamDef
BSW Description		
ID to be sent on the TCP/IP connection if the PDU header option is enabled.		
Template Description		
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.headerId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00120]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		BSW Type
SoAdTxRoutingGroupRef		EcucReferenceDef
BSW Description		
Reference to the routing group.		
Template Description		
Group of Pdus that can be activated or deactivated for transmission over a socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup		
Mapping Rule		Mapping Type
If the PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPdulIdentifier that in turn has a relationship to a PduActivationRoutingGroup then this reference shall point to the SoAdRoutingGroup that is derived from the PduActivationRoutingGroup.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00123]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	

BSW Parameter		BSW Type	
SoAdTxSocketConnOrSocketConnBundleRef		EcucChoiceReferenceDef	
BSW Description			
Choice Reference to a SocketConnection or to a SocketConnectionGroup on which the PDU is to be sent on. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API.			
Template Description			
StaticSocketConnection: Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.			
ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both).			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.localUnicastAddress			
Mapping Rule			Mapping Type
<p>SOME/IP Communication: The PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPdulIdentifier that in turn has a relationship to a PduActivationRouting Group. The PduActivationRoutingGroup is aggregated by a ProvidedService Instance (directly or via the EventHandler). The ProvidedServiceInstance points to a localUnicastAddress. For this localUnicastAddress a SocketConnection Group was created that shall be referenced here.</p> <p>StaticSocketConnection: The PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPdulIdentifier that in turn has a relationship to a StaticSocketConnection. The StaticSocketConnection is aggregated by a SocketAddress for which the SocketConnectionGroup was created that shall be referenced here.</p>			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_SoAd_00034]	

BSW Module		BSW Context	
SoAd		SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		BSW Type	
SoAdTxUdpTriggerMode		EcucEnumerationParamDef	
BSW Description			
Specifies whether a PDU triggers the transmission of the nPduUdpTxBuffer. If this parameter is set to TRIGGER_NEVER, SoAd shall use an nPduUdpTxBuffer for the related socket connection. nPduUdpTxBuffer can only be used for upper layers with IF API, i.e. this parameter shall only be set to TRIGGER_NEVER if all upper layers belonging to the related socket connection have SoAdTxUpperLayerType set to "IF". This parameter is only relevant for UDP connections.			
Template Description			
Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduCollectionTrigger			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	

valid	[ECUC_SoAd_00136]
-------	-------------------

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter	BSW Type	
SoAdTxUdpTriggerTimeout	EcucFloatParamDef	
BSW Description		
Specifies the timeout in [s] the nPduUdpTxBuffer shall be transmitted at the latest after this PDU is put into the buffer. This optional parameter is only relevant if SoAdTxUdpTriggerMode is TRIGGER_NEVER.		
Template Description		
Defines the timeout in seconds the PDU collection shall be transmitted at the latest after this PDU has been put into the buffer.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduCollectionPduTimeout		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00150]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter	BSW Type	
SoAdTxPduCollectionSemantics	EcucEnumerationParamDef	
BSW Description		
Specifies if this PDU shall be collected using a queued or last-is-best semantics. This parameter is only relevant if the PDU collection feature is enabled. Shall only be set to SOAD_COLLECT_LAST_IS_BEST if the related upper layer is configured with SoAdIfTriggerTransmit set to TRUE.		
Template Description		
Specifies if the referenced PduTriggering shall be collected using a queued (i.e. all PDU instances) or last-is-best (i.e. only the last PDU instance) semantics. If this attribute is not present the behavior of "queued" is assumed.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduCollectionSemantics		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00160]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdTxPduCollectionSemantics	
BSW Parameter	BSW Type	
SOAD_COLLECT_LAST_IS_BEST	EcucEnumerationLiteralDef	
BSW Description		
The PDU data will be fetched via <Up>_[SoAd][If]TriggerTransmit just before the transmission executes.		
Template Description		
Only the latest PDU instances are transmitted.		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduCollectionSemanticsEnum.lastIsBest	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdTxPduCollectionSemantics
BSW Parameter	BSW Type
SOAD_COLLECT_QUEUED	EcucEnumerationLiteralDef
BSW Description	
The PDU data will instantly be stored in the context of the SoAd_IfTransmit API.	
Template Description	
All instances of PDUs are transmitted.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduCollectionSemanticsEnum.queued	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	BSW Type
SoAdTxPdulId	EcucIntegerParamDef
BSW Description	
Tx PDU ID of the PDU coming from the PDU Router.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00031]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	BSW Type
SoAdTxPduRef	EcucReferenceDef
BSW Description	
Reference to the global PDU structure	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduTriggering	
Mapping Rule	Mapping Type

This SoAdTxPduRef reference shall be derived from the PduTriggering that is referenced by the SoConIPduIdentifier.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00030]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter	BSW Type	
SoAdTxUpperLayerType	EcucEnumerationParamDef	
BSW Description		
Specifies the upper layer interface type (must be "IF" in case of multiple PduRoutes).		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu		
Mapping Rule	Mapping Type	
The SoAdTxUpperLayerType parameter can be derived from the actual type of the PDU: DcmIPdu -> "Tp" UUDT DcmIPdu: "If" (according to [SWS_Dcm_01101]) ISignalIPdu -> "If" if the PduTriggering referring to the ISignalIPdu is NOT referenced by a TpConnection, "Tp" if the PduTriggering referring to the ISignalIPdu is referenced by a TpConnection. NmPdu -> "If" GeneralPurposePdu with category SD -> "If" GeneralPurposePdu with category DoIP -> "If" for UDP, "Tp" for TCP GeneralPurposePdu with category GLOBAL_TIME -> "If" GeneralPurposeIPdu with category = XCP -> "If" UserDefinedIPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP UserDefinedPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP MultiplexedIPdu: "If" ContainerIPdu: "If" SecuredIPdu: "If" (see limitation in AUTOSAR_SWS_SecureOnboardCommunication)	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00118]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig	
BSW Parameter	BSW Type	
SoAdRoutingGroup	EcucParamConfContainerDef	
BSW Description		
Each container describes a specific routing group which can be enabled or disabled. A routing group consists of PDUs. Routing of PDUs can either be forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a SoAdPduRoute or the other way around specified by a SoAdSocketRoute.		
Template Description		
Group of Pdus that can be activated or deactivated for transmission over a socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup		
Mapping Rule	Mapping Type	

The SoAdRoutingGroup container shall be created for every PduActivationRoutingGroup element that is available in the Ecu Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00109]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup	
BSW Parameter	BSW Type	
SoAdRoutingGroupId	EcucIntegerParamDef	
BSW Description		
Unique ID of Routing Group		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00121]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup	
BSW Parameter	BSW Type	
SoAdRoutingGroupsEnabledAtInit	EcucBooleanParamDef	
BSW Description		
If set to true this routing group will be enabled after initializing the SoAd module (i.e. enabled in the SoAd_Init function).		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00122]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup	
BSW Parameter	BSW Type	
SoAdRoutingGroupTxTriggerable	EcucBooleanParamDef	
BSW Description		
Specifies if the If-TxPDUs related to the PduRouteDest containers referenced by this routing group can be triggered via SoAd_IfRoutingGroupTransmit (TRUE) or not (FALSE).		
Template Description		
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	Mapping Type
The SoAdRoutingGroupTxTriggerable parameter shall be derived from the event GroupControlType enumeration. If triggerUnicast or activationAndTriggerUnicast is set then this parameter shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00146]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig
BSW Parameter	BSW Type
SoAdSocketConnectionGroup	EcucParamConfContainerDef
BSW Description	
Specifies the configuration of a socket connection group, i.e. specifies the socket connections belonging to the group and the parameters which are common for all socket connections of the group. A socket connection specifies how data can be received and transmitted via a TCP or UDP socket.	
Template Description	
<p>ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both).</p> <p>ConsumedServiceInstance.localUnicastAddress: The local address over which the CSI is consumed (udp, tcp or both).</p> <p>StaticSocketConnection: Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.</p>	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.localUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection	
Mapping Rule	Mapping Type

<p>Unicast SocketConnectionGroup rules:</p> <ul style="list-style-type: none"> - SOME/IP Communication without configured remoteUnicastAddress: Collect all ProvidedServiceInstances and ConsumedServiceInstances that are pointing to the same localUnicastAddress of the regarded Ecu and create a SoAdConnectionGroup for the referenced SocketAddress. Each relevant Socket Address aggregates the referenced ApplicationEndpoint and points with the connector reference to the Ecu for which the Ecu Configuration is created. - SOME/IP Communication with configured remoteUnicastAddress: Collect all ProvidedServiceInstances and ConsumedServiceInstances that are pointing to the same localUnicastAddress and to the same remoteUnicastAddress and create a SoAdConnectionGroup for the referenced SocketAddress. Each relevant SocketAddress aggregates the referenced ApplicationEndpoint that is referenced by the localUnicastAddress and points with the connector reference to the Ecu for which the Ecu Configuration is created. - StaticSocketConnection: Create a SocketConnectionGroup for each StaticSocketConnection that is aggregated by an ApplicationEndpoint of the regarded Ecu or points with the remoteUnicastAddress to an ApplicationEndpoint of the regarded Ecu. <p>Multicast SocketConnectionGroup rules:</p> <ul style="list-style-type: none"> - Server side: If a ProvidedServiceInstance points with the localUnicastAddress to an ApplicationEndpoint that in turn is aggregated by a SocketAddress that refers the regarded EcuInstance and this ProvidedServiceInstance contains an Event Handler that contains an eventMulticastRef then a multicast SocketConnection Group shall be created. The local Address of the SocketConnectionGroup shall be the address that is derived from the ProvidedServiceInstance.localUnicast reference. The remote Address of the SocketConnectionGroup shall be the multicastAddress that is derived from EventHandler.eventMulticastAddress. - Client side: If a ConsumedServiceInstance points with the localUnicastAddress to an ApplicationEndpoint that in turn is aggregated by a SocketAddress that refers the regarded EcuInstance and this ConsumedServiceInstance contains a ConsumedEventGroup that contains an eventMulticastRef then a multicast SocketConnectionGroup shall be created. The local Address of the SocketConnectionGroup shall be derived from the ApplicationEndpoint that is referenced by the eventMulticastAddress reference. The remote Address shall be derived from the ApplicationEndpoint that is referenced by ProvidedServiceInstance.localUnicastAddress. 	full				
<table border="1"> <thead> <tr> <th>Mapping Status</th> <th>ECUC Parameter ID</th> </tr> </thead> <tbody> <tr> <td>valid</td> <td>[ECUC_SoAd_00130]</td> </tr> </tbody> </table>	Mapping Status	ECUC Parameter ID	valid	[ECUC_SoAd_00130]	
Mapping Status	ECUC Parameter ID				
valid	[ECUC_SoAd_00130]				

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdPduHeaderEnable	EcucBooleanParamDef
BSW Description	
Enables the transmission of the PDU header (ID, length) on this socket connection. TRUE: add SoAd PDU header before PDU data FALSE: No SoAd PDU header is used	

Template Description	
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.headerId	
Mapping Rule	Mapping Type
<p>If a SoConIPdulIdentifier element has a headerId set to a value and this SoConIPdulIdentifier is referenced by a PduActivationGroup that is aggregated by a relevant ProvidedServiceInstance/ConsumedServiceInstance/EventHandler/ConsumedEventGroup then this parameter shall be set to true.</p> <p>If a StaticSocketConnection refers to a SoConIPdulIdentifier element that has a headerId set to a value then this parameter shall be set to true.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00131]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketAutomaticSoConSetup	EcucBooleanParamDef
BSW Description	
Specifies if the setup of the socket connection shall be done automatically (TRUE) or manually (FALSE) via SoAd_OpenSoCon() and SoAd_CloseSoCon().	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00110]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketConnection	EcucParamConfContainerDef
BSW Description	
<p>Specifies the socket connection (Id and remote address information).</p> <p>Note: Parameters which are common to all socket connections of a socket connection group are specified directly at the group.</p>	
Template Description	
<p>ApplicationEndpoint.maxNumberOfConnections: This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.</p> <p>StaticSocketConnection: Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.</p>	
M2 Parameter	
<p>SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint.maxNumberOfConnections, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection</p>	

Mapping Rule		Mapping Type
<p>SOME/IP communication without configured remoteUnicastAddress: The SocketConnectionBundle is created for an ApplicationEndpoint aggregated by a SocketAddress that contains a connector reference to the regarded Ecu. The relevant ApplicationEndpoint is referenced by the localUnicastAddress reference from ProvidedServiceInstances/ConsumedServiceInstances. The relevant ApplicationEndpoint has the attribute maxNumberOfConnections that defines how many SocketConnections inside the SocketConnectionBundle need to be created. If this attribute is not set then only a single SocketConnection shall be created. The remotePort of each created SocketConnection shall be set to 0 and the IP Address to ANY.</p> <p>SOME/IP communication with configured remoteUnicastAddress: The SocketConnectionBundle is created for an ApplicationEndpoint aggregated by a SocketAddress that contains a connector reference to the regarded Ecu. The relevant ApplicationEndpoint is referenced by the localUnicastAddress reference from ProvidedServiceInstances/ConsumedServiceInstances. For each ProvidedServiceInstance/ConsumedServiceInstance that has a remoteUnicastAddress defined one single SocketConnection shall be defined. he remotePort the IP Address shall be derived from the ApplicationEndpoint/NetworkEndpoint combination that is referenced by the remoteUnicastAddress</p> <p>StaticSocketConnection: If the SocketConnectionBundle was created from a StaticSocketConnection then this SocketConnectionBundle shall always contain exactly one Socket Connection.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00009]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection	
BSW Parameter		BSW Type
SoAdSocketId		EcucIntegerParamDef
BSW Description		
Socket connection identifier used as SoConId in the interaction with upper layers.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00016]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection	
BSW Parameter		BSW Type
SoAdSocketRemoteAddress		EcucParamConfContainerDef
BSW Description		

Subcontainer of SoAdSocketConnection to specify the remote address (IP address and port) for a socket connection. If SoAdSocketRemoteAddress is not specified the remote address has to be set by the upper layer via SoAd_SetRemoteAddr().	
Template Description	
ConsumedServiceInstance.remoteUnicastAddress: This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery.	
ProvidedServiceInstance.remoteUnicastAddress: This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.	
StaticSocketConnection.remoteAddress: RemoteAddress of the static SocketConnection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.remoteAddress	
Mapping Rule	Mapping Type
SOME/IP Communication without configured remoteUnicastAddress: If the remoteUnicastAddress is not set it means that the remoteUnicastAddress is derived via Service Discovery at runtime. In this case the remote Port shall be set to 0 and the IP Address to ANY.	full
SOME/IP Communication with configured remoteUnicastAddress: If the remoteUnicastAddress is used the remoteAddressPort and the remote IP Address shall be derived from the referenced ApplicationEndpoint.	
StaticSocketConnection: In case of the StaticSocketConnection the remoteAddressPort and IP Address shall be derived from the ApplicationEndpoint that is aggregated by the Socket Address that in turn is referenced by the StaticSocketConnection.remoteAddress.	
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00113]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemoteAddress
BSW Parameter	BSW Type
SoAdSocketRemotelpAddress	EcucStringParamDef
BSW Description	
IP address of remote node. The configured address must be of the same TcpIpDomainType (i.e. IPv4 or IPv6) as the TcpIpLocalAddr referred by SoAdSocketLocalAddressRef . To accept any remote IP address, set SoAdSocketRemotelpAddress to "ANY". See message acceptance policy for more details.	
Template Description	

<p>ProvidedServiceInstance.remoteUnicastAddress: This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.</p>	
<p>ConsumedServiceInstance.remoteUnicastAddress: This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery.</p>	
<p>StaticSocketConnection.remoteAddress: RemoteAddress of the static SocketConnection.</p>	
<p>M2 Parameter</p>	
<p>SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.remoteAddress</p>	
<p>Mapping Rule</p>	
<p>SOME/IP Communication without configured remoteUnicastAddress: If the remoteUnicastAddress is not used it means that the remoteAddress is derived via ServiceDiscovery at runtime. In this case this attribute shall be set to ANY.</p> <p>SOME/IP Communication with configured remoteUnicastAddress: If the remoteUnicastAddress is used the remoteAddress IP Address shall be derived from the referenced ApplicationEndpoint that in turn refers a Network Endpoint that contains the searched IP Address.</p> <p>StaticSocketConnection: In case of the StaticSocketConnection the remoteAddress shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn is referenced by the StaticSocketConnection.remoteAddress. The Application Endpoint refers to a NetworkEndpoint that contains the searched IP Address.</p>	<p>Mapping Type</p> <p>full</p>
<p>Mapping Status</p>	
<p>valid</p>	<p>ECUC Parameter ID</p> <p>[ECUC_SoAd_00019]</p>

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemoteAddress
BSW Parameter	BSW Type
SoAdSocketRemotePort	EcucIntegerParamDef
BSW Description	
<p>Remote UDP or TCP port used for this connection. To accept any remote port, set SoAdSocketRemotePort to 0. See message acceptance policy for more details.</p>	
Template Description	

<p>ProvidedServiceInstance.remoteUnicastAddress: This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.</p>	
<p>ConsumedServiceInstance.remoteUnicastAddress: This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery.</p>	
<p>StaticSocketConnection.remoteAddress: RemoteAddress of the static SocketConnection.</p>	
<p>M2 Parameter</p> <p>SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.remoteAddress</p>	
Mapping Rule	Mapping Type
<p>SOME/IP Communication without configured remoteUnicastAddress: If the remoteUnicastAddress is not used it means that the remoteAddress is derived via ServiceDiscovery at runtime. In this case this attribute shall be set to 0.</p> <p>SOME/IP Communication with configured remoteUnicastAddress: If the remoteUnicastAddress is used the remoteAddressPort shall be derived from the referenced ApplicationEndpoint.</p> <p>StaticSocketConnection: In case of the StaticSocketConnection the remoteAddressPort shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn is referenced by the StaticSocketConnection.remoteAddress.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00020]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketDifferentiatedServicesField	EcucIntegerParamDef
BSW Description	
The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.	
Template Description	
The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.differentiatedServiceField	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00158]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		BSW Type
SoAdSocketFlowLabel		EcucIntegerParamDef
BSW Description		
The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.		
Template Description		
The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.flowLabel		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00157]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		BSW Type
SoAdSocketFramePriority		EcucIntegerParamDef
BSW Description		
Specifies the priority of the Ethernet frame. If IEEE 802.1Q VLAN Tags are used, the specified priority will be used in the VLAN Tag PCP field. If this optional parameter is not available the default priority specified in the Tcplp module is used.		
Template Description		
<p>VlanMembership.defaultPriority: Standard output-priority outgoing Frames will be tagged with.</p> <p>Defines the priority that received frames are assigned together with the VLAN Id (default-Vlan). The values from 0 (best effort) to 7 (highest) are allowed.</p> <p>In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.</p> <p>NetworkEndpoint: The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).</p> <p>ApplicationEndpoint.priority: Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.</p> <p>ProvidedServiceInstance.priority: Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.</p> <p>ConsumedEventGroup.priority: Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.</p>		
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.defaultPriority, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpoint SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint.priority SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.priority, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedEventGroup.priority	
Mapping Rule	Mapping Type
<p>It shall be considered that the priority in the SoAd is defined only once per SocketConnectionGroup. The SocketConnections in the system description shall be created adequate.</p> <p>The priority in the system description can be defined at the ConsumedEventGroup, ProvidedServiceInstance, at the ApplicationEndpoint and at the NetworkEndpoint. A default priority can be set at CouplingPort.VlanMembership. The priority defined on the ProvidedServiceInstance or ConsumedEventGroup overrides the priority defined on the ApplicationEndpoint. The priority on the ApplicationEndpoint overrides the priority on the NetworkEndpoint. The priority on the NetworkEndpoint overrides the default priority.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00138]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketIpAddrAssignmentChgNotification	EcucBooleanParamDef
BSW Description	
Specifies if the local IP address assignment change notification callback function of the upper layer shall be called if the assignment of the local IP address used by this socket connection changes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00112]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketLocalAddressRef	EcucReferenceDef
BSW Description	
Local IP address and interface used for this connection.	
Template Description	

ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both).	
ConsumedServiceInstance.remoteUnicastAddress: This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery.	
SocketAddress.staticSocketConnection: Definition of a static SocketConnection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.localUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.remoteUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.staticSocketConnection	
Mapping Rule	Mapping Type
The SoAdSocketLocalAddressRef shall be derived from the attributes IPv4Configuration.ipv4Address or IPv6Configuration.ipv6Address from the NetworkEndpoint that is referenced by the ApplicationEndpoint that is aggregated by the SocketAddress referenced from the ProvidedServiceInstance, ConsumedServiceInstance or that aggregates the StaticSocketConnection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00017]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketLocalPort	EcucIntegerParamDef
BSW Description	
Local UDP or TCP port used for this connection. If this parameter set to 0 SoAd requests TcpIp to select an ephemeral port.	
Template Description	
ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both).	
ConsumedServiceInstance.localUnicastAddress: The local address over which the CSI is consumed (udp, tcp or both).	
SocketAddress.staticSocketConnection: Definition of a static SocketConnection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ProvidedServiceInstance.localUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress, SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.staticSocketConnection	
Mapping Rule	Mapping Type

<p>The SoAdSocketLocalPort shall be derived from the value of the attribute Tp Port.portNumber in the ApplicationEndpoint referenced from ProvidedService Instance/ConsumedServiceInstance in the role localUnicastAddress.</p> <p>In case of a StaticSocketConnection the local port shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn aggregates the StaticSocketConnection.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00018]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter	BSW Type	
SoAdSocketMsgAcceptanceFilterEnabled	EcucBooleanParamDef	
BSW Description		
<p>Specifies if the message acceptance filter is enabled (TRUE) or not (FALSE). Note: if a wildcard is used in SoAdSocketRemoteAddress AND SoAdSocketUdpListenOnly is FALSE, this parameter must be TRUE. Note: if multiple SoAdSocketConnections are configured for one SoAdSocketConnectionGroup, this parameter must be TRUE.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00137]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter	BSW Type	
SoAdSocketPathMTUEnable	EcucBooleanParamDef	
BSW Description		
<p>Specifies if path MTU discovery shall be performed for this connection. If this optional parameter is not available the default behavior configured for the controller in the Tcplp module via the parameter TcplpV4PathMtuEnabled or TcplpV6PathMtuEnabled is applied.</p>		
Template Description		
Defines whether the Path MTU Discovery shall be performed for the related socket.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.pathMtuDiscovery Enabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00156]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter	BSW Type	
SoAdSocketProtocol	EcucChoiceContainerDef	

BSW Description	
Specifies the transport protocol and transport protocol specific parameters used for the socket connections of the socket connection group.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00139]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol
BSW Parameter	BSW Type
SoAdSocketTcp	EcucParamConfContainerDef
BSW Description	
Specifies that TCP is used as transport protocol for the socket connection group and parameters only related to TCP socket connections.	
Template Description	
An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint	
Mapping Rule	Mapping Type
<p>Server Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ProvidedServiceInstance in the role localUnicast Address. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.</p> <p>Client Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ConsumedServiceInstance in the role localUnicast Address. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00141]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	BSW Type
SoAdSocketTCPOptionFilterRef	EcucReferenceDef
BSW Description	
Specifies which TCP option filter shall be applied on the related socket.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local

Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00155]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpImmediateTpTxConfirmation		EcucBooleanParamDef
BSW Description		
If set to FALSE, SoAd notifies the TP upper layer via transmit confirmation after a Tcp Ack has been received. If set to TRUE, SoAd notifies the TP upper layer via transmit confirmation immediately after transmit has been accepted by TcpIp.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00147]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpInitiate		EcucBooleanParamDef
BSW Description		
Specifies the initiator for this TCP connection. It will not be defined for UDP sockets. TRUE: This TCP connection is initiated by this module. FALSE: This TCP connection is to be initiated in the listen mode.		
Template Description		
Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.tcpRole		
Mapping Rule		Mapping Type
<p>SOME/IP Communication:</p> <ul style="list-style-type: none"> - Server Ecu: SoAdSocketTcpInitiate can be set to false here since Servers do not initiate Tcp connections by themselves. - Client Ecu: SoAdSocketTcpInitiate can be set to true if TcpTp is configured and the Ecu is in the client role. <p>StaticSocketConnection: If StaticSocketConnection.tcpRole = connect then this attribute shall be set to TRUE. If StaticSocketConnection.tcpRole = listen then this attribute shall be set to TRUE.</p>		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00022]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpKeepAlive		EcucBooleanParamDef
BSW Description		
Specifies to use the keep-alive mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will use the keep-alive mechanism. FALSE: This TCP connection will not use the keep-alive mechanism. Note: This parameter must not be set to TRUE if TcpIpTcpKeepAliveEnabled is set to FALSE.		
Template Description		
Indicates if Keep-Alive messages are send.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAlives		
Mapping Rule		Mapping Type
1:1 mapping		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00148]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpKeepAliveInterval		EcucFloatParamDef
BSW Description		
Specifies the interval in seconds between subsequent keepalive probes.		
Template Description		
Specifies the interval in seconds between subsequent keepalive probes.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAliveInterval		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00152]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpKeepAliveProbesMax		EcucIntegerParamDef
BSW Description		
Maximum number of times that TCP retransmits an individual data segment before aborting the connection.		
Template Description		
Maximum number of times that TCP retransmits an individual data segment before aborting the connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAliveProbesMax		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00151]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpKeepAliveTime		EcucFloatParamDef
BSW Description		
Specifies the time in seconds between the last data packet sent and the first keepalive probe.		
Template Description		
Specifies the time in seconds between the last data packet sent and the first keepalive probe.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAliveTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00153]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpNoDelay		EcucBooleanParamDef
BSW Description		
Specifies not to use the congestion control mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will NOT use congestion control. FALSE: This TCP connection will use congestion control. If the optional parameter is not enabled, the default behavior configured for TcpIp via the parameter TcpIpTcpNagleEnabled is applied. Note: This parameter must not be set to FALSE if TcpIpTcpNagleEnabled is set to FALSE.		
Template Description		
Indicates if Nagle's Algorithm is used.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.naglesAlgorithm		
Mapping Rule		Mapping Type
If TcpTp.naglesAlgorithm in the System Extract is set to true then SoAdSocketTcpNoDelay shall be set to false. If TcpTp.naglesAlgorithm in is set to false then SoAdSocketTcpNoDelay shall be set to true. If TcpTp.naglesAlgorithm in the System Extract is not defined then SoAdSocketTcpNoDelay shall not be set.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00023]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		BSW Type
SoAdSocketTcpTlsConnectionRef		EcucReferenceDef
BSW Description		

If set the TCP socket is assigned to a TLS connection. The SoAd need to call Tcplp_ChangeParameter with the reference to the TLS connection as the parameter.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00163]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	BSW Type
SoAdSocketTcpTxQuota	EcucIntegerParamDef
BSW Description	
Specifies the maximum amount of bytes (PDU data provided by the upper layer and PDU Header if used) the SoAd may queue for transmission via TCP at the Tcplp module for each socket connection of this socket connection group.	
Rationale: prohibits that a socket connection consumes all available transmit buffers at the Tcplp and blocks transmissions via other socket connections. If the optional parameter is not enabled, the amount of data is not limited.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00142]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol
BSW Parameter	BSW Type
SoAdSocketUdp	EcucParamConfContainerDef
BSW Description	
Specifies that UDP is used as transport protocol for the socket connection group and parameters only related to UDP socket connections.	
Template Description	
An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint	
Mapping Rule	Mapping Type

<p>Server Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ProvidedServiceInstance in the role localUnicast Address. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.</p> <p>Client Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ConsumedServiceInstance in the role localUnicast Address. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.</p>		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00140]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter	BSW Type	
SoAdSocketUdpAliveSupervisionTimeout	EcucFloatParamDef	
BSW Description		
Specifies the time in [s] a UDP socket connection remains in the mode SOAD_SOCON_ONLINE after the latest reception of a frame from the remote peer specified by the remote address. If this optional parameter is not enabled UDP Alive Supervision is deactivated for the related socket connection group.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00149]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter	BSW Type	
SoAdSocketUdpChecksumEnabled	EcucBooleanParamDef	
BSW Description		
Specifies if UDP checksum calculation shall be enabled (TRUE) or skipped (FALSE) on the related socket. FALSE implies that the upper layer of the socket connection is either capable to handle malformed messages or applies a checksum mechanism itself.		
Template Description		
Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksumDisabled) on the related socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.udpChecksumHandling		
Mapping Rule		Mapping Type
If udpChecksumHandling.udpChecksumEnabled is set the value shall be TRUE; if udpChecksumHandling.udpChecksumDisabled is set the value shall be FALSE;		full
Mapping Status	ECUC Parameter ID	

valid	[ECUC_SoAd_00159]
-------	-------------------

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		BSW Type
SoAdSocketUdpListenOnly		EcucBooleanParamDef
BSW Description		
Specifies if the socket connection group is only used for reception (TRUE) or used for both reception and transmission (FALSE).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00024]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		BSW Type
SoAdSocketUdpStrictHeaderLenCheckEnabled		EcucBooleanParamDef
BSW Description		
Specifies if UDP messages shall be dropped (TRUE) if the length of all contained PDUs does not match the length of the whole message or not (FALSE). Shall only be set to TRUE if SoAdPduHeaderEnable is also set to TRUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00154]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		BSW Type
SoAdSocketUdpTriggerTimeout		EcucFloatParamDef
BSW Description		
Specifies the timeout in [s] a nPduUdpTxBuffer is waiting for a PDU with TriggerMode = TRIGGER_ALWAYS, i.e. when the timeout expires the nPduUdpTxBuffer is transmitted. Timer is reset after each UDP transmission. This optional parameter is only relevant if a nPduUdpTxBuffer is used.		
Template Description		
Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.pduCollectionTimeout		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00133]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter	BSW Type	
SoAdSocketnPduUdpTxBufferMin	EcucIntegerParamDef	
BSW Description		
<p>Specifies the amount of data in bytes (PDU data provided by the upper layer and PDU Header if used) the SoAd shall be able to buffer for data transmission via this socket connection in case the UDP message shall be buffered for transmission of multiple PDUs per UDP.</p> <p>Note: in case of a UDP socket and an upper layer with TP API is configured, the required buffer size can be determined automatically. This optional parameter is only relevant if a nPduUdpTxBuffer is used.</p>		
Template Description		
Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.pduCollectionMaxBufferSize		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00135]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter	BSW Type	
SoAdSocketSoConModeChgNotifUpperLayerRef	EcucReferenceDef	
BSW Description		
Reference to an additional upper layer that shall receive socket connection state changes (although it is not a direct upper layer of the socket connection).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SoAd_00161]	

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter	BSW Type	
SoAdSocketSoConModeChgNotification	EcucBooleanParamDef	
BSW Description		

Specifies if the SoCon mode change notification callback function of the upper layer shall be called in case of SoCon mode change.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00111]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	BSW Type
SoAdSocketTpRxBufferMin	EcucIntegerParamDef
BSW Description	
Specifies the amount of data in bytes (PDU data for the upper layer and PDU Header if used) the SoAd shall at least be able to buffer for data reception via each socket connection of the socket connection group and using an upper layer with TP.	
Note: in case of a TCP socket where PduHeaderMode is used and an upper layer with IF-API, the required buffer size can be determined automatically.	
Template Description	
Minimum size of the TCP receive window in byte.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.receiveWindowMin	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00134]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig
BSW Parameter	BSW Type
SoAdSocketRoute	EcucParamConfContainerDef
BSW Description	
Describes the path of a PDU from a socket in the TCP/IP stack to an upper layer of the SoAd after reception in the TCP/IP Stack.	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduTriggering	
Mapping Rule	Mapping Type
The SoAdSocketRoute container shall be created for every PduTriggering that is received by the regarded ECU. The information whether the Pdu is received or transmitted over a SocketConnection shall be derived from the PduTriggering element. The PduTriggering element contains references to IPduPorts of an EcucInstance. The IPduPort element contains a communicationDirection.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00008]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter	BSW Type	
SoAdRxPduHeaderId	EcucIntegerParamDef	
BSW Description		
ID contained in the packet received on the TCP/IP connection if the PDU header option is enabled.		
Template Description		
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.headerId		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00036]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter	BSW Type	
SoAdRxSocketConnOrSocketConnBundleRef	EcucChoiceReferenceDef	
BSW Description		
Choice Reference to a SocketConnection or to a SocketConnectionGroup on which the PDU was received. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API.		
Template Description		
SocketAddress.staticSocketConnection: Definition of a static SocketConnection.		
ConsumedServiceInstance.localUnicastAddress: The local address over which the CSI is consumed (udp, tcp or both).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.staticSocketConnection SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::ConsumedServiceInstance.localUnicastAddress		
Mapping Rule		Mapping Type
<p>SOME/IP Communication:</p> <p>The PduTriggering for which the SoAdSocketRoute is created is referenced by a SoConIPdulIdentifier that in turn has a relationship to a PduActivationRoutingGroup. The PduActivationRoutingGroup is aggregated by a ConsumedServiceInstance (directly or via the ConsumedEventGroup). The ConsumedServiceInstance points to a localUnicastAddress. For this localUnicastAddress a SocketConnectionGroup was created that shall be referenced here.</p> <p>StaticSocketConnection:</p> <p>The PduTriggering for which the SoAdSocketRoute is created is referenced by a SoConIPdulIdentifier that in turn has a relationship to a StaticSocketConnection. The StaticSocketConnection is aggregated by a SocketAddress for which the SocketConnectionGroup was created that shall be referenced here.</p>		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00035]

BSW Module	BSW Context
------------	-------------

SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter		BSW Type
SoAdSocketRouteDest		EcucParamConfContainerDef
BSW Description		
Describes the upper layer destination PDU for a message received on a TcpIp socket. This PDU can produce meta data items of type SOCKET_CONNECTION_ID_16.		
Multiple socket route destinations in the SoAdSocketRoute can only be used for upper layers of interface type (IF) and only for SoAdSocketRoute referring a SocketConnectionGroup. In this case SoAdRoutingGroups shall be used to map each SoAdSocketRouteDest uniquely to different socket connections of the SocketConnectionGroup.		
Template Description		
Identification of Pdu content on a socket connection. This Identifier is required in case that multiple Pdus are transmitted over the same socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier		
Mapping Rule		Mapping Type
The SoAdSocketRouteDest container shall always be created for a SoAdSocket Route.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00114]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		BSW Type
SoAdRxPdul		EcucIntegerParamDef
BSW Description		
This unique identifier is used for a receive cancellation request from an upper layer of the SoAd.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00116]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		BSW Type
SoAdRxPduRef		EcucReferenceDef
BSW Description		
Reference to the global PDU structure		
Template Description		
Reference to a Pdu that is transmitted over a socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduTriggering		
Mapping Rule		Mapping Type
The SoAdRxPduRef reference shall be derived from the PduTriggering that is referenced by the SoConIPdulIdentifier.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00038]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		BSW Type
SoAdRxRoutingGroupRef		EcucReferenceDef
BSW Description		
Reference to the routing group. Mandatory if the parent SoAdSocketRoute contains more than one SoAdSocketRouteDest."		
Template Description		
Group of Pdus that can be activated or deactivated for transmission over a socket connection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::PduActivationRoutingGroup		
Mapping Rule		Mapping Type
The SoAdRxRoutingGroupRef reference to SoAdRoutingGroups shall be derived from the PduActivationRoutingGroup that refers the SoConIPduIdentifier.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00117]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		BSW Type
SoAdRxUpperLayerType		EcucEnumerationParamDef
BSW Description		
Specifies the upper layer interface type (must be "IF" in case of multiple RxPdus).		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPdu		
Mapping Rule		Mapping Type
The SoAdRxUpperLayerType parameter can be derived from the actual type of the PDU: DcmIPdu -> "Tp" UUDT DcmIPdu: "If" (according to [SWS_Dcm_01101]) ISignalIPdu -> "If" if ComIPduType=NORMAL, "Tp" if ComIPduType=TP NmPdu -> "If" GeneralPurposePdu with category SD -> "If" GeneralPurposePdu with category DoIP -> "If" for UDP, "Tp" for TCP GeneralPurposePdu with category GLOBAL_TIME -> "If" GeneralPurposeIPdu with category = XCP -> "If" UserDefinedIPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP UserDefinedPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP MultiplexedIPdu: "If" ContainerIPdu: "If" SecuredIPdu: "If" (see limitation in AUTOSAR_SWS_SecureOnboardCommunication)		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_SoAd_00115]

BSW Module	BSW Context	
SoAd	SoAd	
BSW Parameter		BSW Type
SoAdGeneral		EcucParamConfContainerDef
BSW Description		

This container contains all global configuration parameters of SoAd.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00003]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00002]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdGetAndResetMeasurementDataApi	EcucBooleanParamDef
BSW Description	
Enables / Disables the Get and Reset Measurement Data API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00162]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdIPv6AddressEnabled	EcucBooleanParamDef
BSW Description	

Allows for increased memory allocation to store IPv6 addresses.	
TRUE: Enables support for IPv6 addresses FALSE: Only IPv4 addresses are supported	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00039]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Determines the frequency at which the SoAd_MainFunction() is called in [s].	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00062]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdRoutingGroupMax	EcucIntegerParamDef
BSW Description	
Specifies the maximum number of SoAd routing groups. Furthermore it defines the platform type used for RoutingGroupIdType. If SoAdRoutingGroupMax is not greater than 256, a uint8 is used, otherwise a uint16.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00127]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdSoConMax	EcucIntegerParamDef
BSW Description	

Specifies the maximum number of SoAd socket connections.
Furthermore it defines the platform type used for SoAd_SoConIdType. If SoAdSoConMax is not greater than 256, a uint8 is used, otherwise uint16.

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00126]

BSW Module	BSW Context
SoAd	SoAd/SoAdGeneral
BSW Parameter	BSW Type
SoAdVersionInfoApi	EcucBooleanParamDef
BSW Description	
Activates the SoAd_GetVersionInfo() API. TRUE: Enables the SoAd_GetVersionInfo() API. FALSE: SoAd_GetVersionInfo() API is not included.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SoAd_00004]

C.6.6 EthSM

BSW Module	BSW Context
EthSM	EthSM
BSW Parameter	BSW Type
EthSMGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the global parameter of the Ethernet State Manager.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00063]

BSW Module	BSW Context
EthSM	EthSM/EthSMGeneral
BSW Parameter	BSW Type
EthSMDevErrorDetect	EcucBooleanParamDef

BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00065]

BSW Module	BSW Context
EthSM	EthSM/EthSMGeneral
BSW Parameter	BSW Type
EthSMDummyMode	EcucBooleanParamDef
BSW Description	
Disables the API to the EthIf. The API to the ComM is available but the functionality is deactivated. The function calls from the ComM will be answered with the return value E_OK.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00079]

BSW Module	BSW Context
EthSM	EthSM/EthSMGeneral
BSW Parameter	BSW Type
EthSMMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Specifies the period in seconds that the MainFunction has to be triggered with.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00066]

BSW Module	BSW Context
EthSM	EthSM/EthSMGeneral
BSW Parameter	BSW Type
EthSMVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables and disables the version info API.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00092]

BSW Module	BSW Context
EthSM	EthSM
BSW Parameter	BSW Type
EthSMNetwork	EcucParamConfContainerDef
BSW Description	
This container contains the Ethernet network-specific parameters of each Ethernet network. It also contains the reference to combination of controller and transceiver assigned to an Ethernet network.	
Template Description	
The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalChannel	
Mapping Rule	Mapping Type
1:1	full
For each EthernetPhysicalChannel the EcuInstance is connected to, one EthSMNetwork container is created.	
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00067]

BSW Module	BSW Context
EthSM	EthSM/EthSMNetwork
BSW Parameter	BSW Type
EthSMComMNetworkHandleRef	EcucReferenceDef
BSW Description	
Unique handle to identify one certain Ethernet network. Reference to one of the network handles configured for the ComM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00068]

BSW Module	BSW Context
EthSM	EthSM/EthSMNetwork
BSW Parameter	BSW Type
EthSMDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00106]

BSW Module	BSW Context
EthSM	EthSM/EthSMNetwork/EthSMDemEventParameterRefs
BSW Parameter	BSW Type
ETHSM_E_LINK_DOWN	EcucReferenceDef
BSW Description	
Reference to configured DEM event to report bus off errors for this Eth network.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00107]

BSW Module	BSW Context
EthSM	EthSM/EthSMNetwork
BSW Parameter	BSW Type
EthSMEthIfControllerRef	EcucReferenceDef
BSW Description	
Reference to EthIfCtrl container where a ETH controller and transceiver (optional) combination is configured.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSM_00105]

C.6.7 EthTrcv

BSW Module	BSW Context
EthTrcv	EthTrcv
BSW Parameter	BSW Type
EthTrcvConfigSet	EcucParamConfContainerDef

BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR EthTrcv module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00016]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet
BSW Parameter	BSW Type
EthTrcvConfig	EcucParamConfContainerDef
BSW Description	
Configuration of the individual transceiver	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00012]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvConfigEcucPartitionRef	EcucReferenceDef
BSW Description	
Maps the Ethernet transceiver configuration to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Ethernet transceiver driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00051]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvConnNeg	EcucEnumerationParamDef
BSW Description	
Specifies the connection negotiation of the Ethernet transceiver link.	
Template Description	

Specifies the connection negotiation of the CouplingPort.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.connectionNegotiationBehavior	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00025]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg
BSW Parameter	BSW Type
TRCV_CONN_NEG_AUTO	EcucEnumerationLiteralDef
BSW Description	
Automatic Negotiation	
Template Description	
Automatic Negotiation	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetConnectionNegotiationEnum.auto	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg
BSW Parameter	BSW Type
TRCV_CONN_NEG_MASTER	EcucEnumerationLiteralDef
BSW Description	
Master	
Template Description	
Master	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetConnectionNegotiationEnum.master	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg
BSW Parameter	BSW Type
TRCV_CONN_NEG_SLAVE	EcucEnumerationLiteralDef
BSW Description	
Slave	
Template Description	
Slave	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetConnectionNegotiation	
Enum.slave	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00017]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvDemEventParameterRefs
BSW Parameter	BSW Type
ETHTRCV_E_ACCESS	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occurred.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00018]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvDuplexMode	EcucEnumerationParamDef
BSW Description	
Specifies the duplex mode of the Ethernet transceiver link if Auto-Negotiation is disabled. This parameter is ignored if Auto-Negotiation is enabled (EthTrcvConnNeg=TRCV_CONN_NEG_AUTO).	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00023]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvIcuChannelRef	EcucReferenceDef
BSW Description	
Reference to the IcuChannel to enable/disable the interrupts for wakeups.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00026]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvIdx	EcucIntegerParamDef
BSW Description	
Specifies the instance ID of the configured transceiver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00013]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvMacLayerSpeed	EcucEnumerationParamDef
BSW Description	
Defines the baud rate of the MAC layer.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_EthTrcv_00044]
-------	----------------------

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter	BSW Type	
EthTrcvMacLayerSubType	EcucEnumerationParamDef	
BSW Description	Defines the MAC layer subtype of a switch port	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00043]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter	BSW Type	
EthTrcvMacLayerType	EcucEnumerationParamDef	
BSW Description	Defines the MAC layer type of the ethernet transceiver.	
Template Description	Specifies the mac layer type of the CouplingPort.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.macLayerType	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00035]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	
BSW Parameter	BSW Type	
TRCV_MAC_LAYER_TYPE_XGMII	EcucEnumerationLiteralDef	
BSW Description	MAC layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
Template Description	Mac layer interface (data) bandwidth class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xGMII	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	

BSW Parameter		BSW Type	
TRCV_MAC_LAYER_TYPE_XMII		EcucEnumerationLiteralDef	
BSW Description			
MAC layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)			
Template Description			
Mac layer interface (data) bandwidth class 100Mbit/s (e.g. RMII, RvMII, SMII, RvMII)			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xMII			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	
BSW Parameter		BSW Type	
TRCV_MAC_LAYER_TYPE_XXGMII		EcucEnumerationLiteralDef	
BSW Description			
MAC layer interface (data) bandwidth class 10Gbit/s			
Template Description			
Mac layer interface (data) bandwidth class 10Gbit/s			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetMacLayerTypeEnum.xXGMII			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		BSW Type	
EthTrcvMgmtInterface		EcucChoiceContainerDef	
BSW Description			
The choice container allow to configure either the EthTrcv is accessed by a MII interface or Switch interface.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00036]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface	
BSW Parameter		BSW Type	
EthTrcvMiiInterface		EcucParamConfContainerDef	
BSW Description			

This container includes the MII interface configuration between an Ethernet Controller and the Ethernet Transceiver. If this container is configured the EthTrcv shall call Eth_WriteMii / Eth_ReadMii API to access the hardware ethernet transceiver.

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00037]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface/EthTrcvMiiInterface
BSW Parameter	BSW Type
EthTrcvCtrlIdx	EcucIntegerParamDef
BSW Description	
Specifies the controller used for MII access to the transceiver	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00014]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface/EthTrcvMiiInterface
BSW Parameter	BSW Type
EthTrcvMiildx	EcucIntegerParamDef
BSW Description	
Specifies the transceiver index used for MII access to the transceiver.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00038]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface
BSW Parameter	BSW Type
EthTrcvSwitchInterface	EcucParamConfContainerDef
BSW Description	

This container includes the Switch interface configuration between an Ethernet Switch and an Ethernet Transceiver. If this container is configured the EthTrcv shall call EthSwt_WriteTrcvRegister / EthSwt_WriteTrcvRegister API to access the hardware ethernet transceiver.

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00040]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface/EthTrcvSwitch Interface
BSW Parameter	BSW Type
EthTrcvSwitchPortRef	EcucReferenceDef
BSW Description	
Reference to a switch port.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00042]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMgmtInterface/EthTrcvSwitch Interface
BSW Parameter	BSW Type
EthTrcvSwitchRef	EcucReferenceDef
BSW Description	
Reference to a switch configuration container.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00041]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvPhysLayerType	EcucEnumerationParamDef
BSW Description	
Specifies the physical layer type of the Ethernet transceiver link.	

Template Description	
Specifies the physical layer type of the CouplingPort.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort.physicalLayerType	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00024]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	BSW Type
TRCV_PHYS_LAYER_TYPE_1000BASE_T	EcucEnumerationLiteralDef
BSW Description	
physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.	
Template Description	
Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalLayerTypeEnum.1000BASE-T	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	BSW Type
TRCV_PHYS_LAYER_TYPE_1000BASE_T1	EcucEnumerationLiteralDef
BSW Description	
physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	BSW Type
TRCV_PHYS_LAYER_TYPE_100BASE_T1	EcucEnumerationLiteralDef
BSW Description	
physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100BASE-T1 is the IEEE Standardized version of BroadRReach.	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalLayerTypeEnum.100BASE-T1	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	BSW Type
TRCV_PHYS_LAYER_TYPE_100BASE_TX	EcucEnumerationLiteralDef
BSW Description	
physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.	
Template Description	
Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalLayerTypeEnum.100BASE-TX	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvSpeed	EcucEnumerationParamDef
BSW Description	
Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled (EthTrcvConnNeg=TRCV_CONN_NEG_AUTO) this is the maximum speed advertised for Auto-Negotiation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00022]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	BSW Type
EthTrcvWakeUpCallout	EcucFunctionNameDef
BSW Description	
Configuration of the call-out name.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00028]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		BSW Type
EthTrcvWakeupMap		EcucParamConfContainerDef
BSW Description		
Container for the mapping of wake up reasons to wake up sources. At least one container is needed if EthTrcvWakeUpSupport is not ETHTRCV_WAKEUP_NOT_SUPPORTED.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00027]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvWakeupMap	
BSW Parameter		BSW Type
EthTrcvWakeupReason		EcucEnumerationParamDef
BSW Description		
This parameter defines the transceiver wake up reasons.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00033]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvWakeupMap	
BSW Parameter		BSW Type
EthTrcvWakeupSourceRef		EcucReferenceDef
BSW Description		
Configures the wake-up source defined in EcuM.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00029]	

BSW Module		BSW Context	
EthTrcv		EthTrcv	
BSW Parameter		BSW Type	
EthTrcvGeneral		EcucParamConfContainerDef	
BSW Description			
General configuration of Ethernet Transceiver Driver module			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00001]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00003]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvEcucPartitionRef		EcucReferenceDef	
BSW Description			
Maps the Ethernet transceiver driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Ethernet transceiver driver will operate as an independent instance in each of the partitions.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00050]	

BSW Module		BSW Context	
------------	--	-------------	--

EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	
EthTrcvEnableCableDiagnosticApi	EcucBooleanParamDef
BSW Description	
Enable/disable the APIs for cable diagnostic: EthTrcv_RunCableDiagnostic, EthTrcv_GetCableDiagnosticsResult	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthTrcv_00054]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	
EthTrcvGetBaudRateApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_GetBaudRate API	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthTrcv_00010]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	
EthTrcvGetCableDiagnosticsResultApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_GetCableDiagnosticsResult API.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	
valid	ECUC Parameter ID [ECUC_EthTrcv_00052]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	
EthTrcvGetDuplexModeApi	EcucBooleanParamDef
BSW Description	

Enables / Disables EthTrcv_GetDuplexMode API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00011]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvGetLinkStateApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_GetLinkState API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00009]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvGetPhyIdentifierApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_GetPhyIdentifier API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00046]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvGetPhySignalQualityApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_GetPhySignalQuality API.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00045]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type
EthTrcvGetTransceiverModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthTrcv_GetTransceiverMode API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00007]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type
EthTrcvGetTransceiverWakeupModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthTrcv_GetTransceiverWakeupMode API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00031]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type
EthTrcvIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00020]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
Specifies the period of main function EthTrcv_MainFunction in seconds.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00032]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvMaxTrcvSupported		EcucIntegerParamDef	
BSW Description			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00002]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvSetPhyTestModeApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables EthTrcv_SetPhyTestMode API.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTrcv_00047]	

BSW Module		BSW Context	
EthTrcv		EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type	
EthTrcvSetPhyTxModeApi		EcucBooleanParamDef	
BSW Description			

Enables / Disables EthTrcv_SetPhyTxMode API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00048]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvSetTransceiverModeApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_SetTransceiverMode API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00006]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvStartAutoNegotiationApi	EcucBooleanParamDef
BSW Description	
Enables / Disables EthTrcv_StartAutoNegotiation API	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTrcv_00008]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral
BSW Parameter	BSW Type
EthTrcvVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables / Disables version info API	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00004]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type
EthTrcvVersionInfoApiMacro		EcucBooleanParamDef
BSW Description		
Enables / Disables version info API macro implementation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00005]	

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		BSW Type
EthTrcvWakeUpSupport		EcucEnumerationParamDef
BSW Description		
Configures wake-up to polling or interrupt or to not used/not supported. In case no wake up is supported by the hardware, the BSWMD pre-configuration shall be set to ETHTRCV_WAKEUP_NOT_SUPPORTED.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTrcv_00030]	

C.6.8 Tcplp

BSW Module	BSW Context	
Tcplp	Tcplp	
BSW Parameter		BSW Type
TcplpConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00003]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig	
BSW Parameter		BSW Type
TcplpCtrl		EcucParamConfContainerDef
BSW Description		
Specifies the EthIf controller used for IP communication.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00021]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl	
BSW Parameter		BSW Type
TcplpDhcpServerConfigRef		EcucReferenceDef
BSW Description		
Reference to a TcplpDhcpServerConfig which shall be used for this controller setting (VLAN).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00195]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl	
BSW Parameter		BSW Type
TcplpEthIfCtrlRef		EcucReferenceDef
BSW Description		
Reference to EthIf controller where the IP address shall be assigned.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00041]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl	
BSW Parameter		BSW Type
TcplpFramePrioDefault		EcucIntegerParamDef
BSW Description		
<p>Specifies the default value for the priority for all outgoing frames. Note: the value can be changed for each socket individually via Tcplp_ChangeParameter() service. If this optional parameter is not available, 0 is used as default priority.</p>		
Template Description		
<p>Standard output-priority outgoing Frames will be tagged with.</p> <p>Defines the priority that received frames are assigned together with the VLAN Id (default-Vlan). The values from 0 (best effort) to 7 (highest) are allowed.</p> <p>In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.defaultPriority		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00081]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl	
BSW Parameter		BSW Type
TcplpSecConfigSetRef		EcucReferenceDef
BSW Description		
Reference to set of SDP entries which shall be used for IPsec.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00315]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl	
BSW Parameter		BSW Type
TcplpVXCtrl		EcucChoiceContainerDef
BSW Description		
Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00094]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl	
BSW Parameter	BSW Type	
TcplpV4Ctrl	EcucParamConfContainerDef	
BSW Description		
Specifies an Internet Protocol version 4 (IPv4) instance.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00166]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl	
BSW Parameter	BSW Type	
TcplpArpConfigRef	EcucReferenceDef	
BSW Description		
Reference to ARP configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00097]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl	
BSW Parameter	BSW Type	
TcplpAutolpConfigRef	EcucReferenceDef	
BSW Description		
Reference to Autolp configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00098]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl	
BSW Parameter	BSW Type	
TcplpDhcpConfigRef	EcucReferenceDef	
BSW Description	Reference to DHCP configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently)	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00100]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl	
BSW Parameter	BSW Type	
TcplpFragmentationConfigRef	EcucReferenceDef	
BSW Description	Reference to Fragmentation configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently)	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00099]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl	
BSW Parameter	BSW Type	
TcplpV4MtuConfig	EcucParamConfContainerDef	
BSW Description	This container specifies the Maximum Transmission Unit parameters for this IPv4 instance.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00209]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl/TcplpV4MtuConfig	
BSW Parameter	BSW Type	

TcplpV4PathMtuEnabled	EcucBooleanParamDef
BSW Description	
If enabled the IPv4 processes incoming ICMPv4 "Packet Too Big" messages and stores a MTU value for each destination address.	
Template Description	
If enabled the IPv4/IPv6 processes incoming ICMP "Packet Too Big" messages and stores a MTU value for each destination address.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pathMtuEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00211]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV4Ctrl/TcplpV4MtuConfig
BSW Parameter	BSW Type
TcplpV4PathMtuTimeout	EcucFloatParamDef
BSW Description	
If this value is >0 the IPv4 will reset the MTU value stored for each destination after n seconds. see [RFC1191 6.3. Purging stale PMTU information] Default: 600 seconds (10 minutes)	
Template Description	
If this value is >0 the IPv4/IPv6 will reset the MTU value stored for each destination after n seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pathMtuTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00210]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl
BSW Parameter	BSW Type
TcplpV6Ctrl	EcucParamConfContainerDef
BSW Description	
Specifies an Internet Protocol version 6 (IPv6) instance.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00096]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl

BSW Parameter		BSW Type	
TcplpV6DhcpConfigRef		EcucReferenceDef	
BSW Description			
Reference to DHCPv6 configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently)			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00101]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl	
BSW Parameter		BSW Type	
TcplpV6FragmentationConfigRef		EcucReferenceDef	
BSW Description			
Reference to IPv6 Fragmentation Configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently)			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00103]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl	
BSW Parameter		BSW Type	
TcplpV6MtuConfig		EcucParamConfContainerDef	
BSW Description			
This container specifies the Maximum Transmission Unit parameters for this IPv6 instance.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00104]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl/TcplpV6MtuConfig	
BSW Parameter		BSW Type	
TcplpV6PathMtuEnabled		EcucBooleanParamDef	
BSW Description			

If enabled the IPv6 processes incoming ICMPv6 "Packet Too Big" messages and stores a MTU value for each destination address.	
See RFC1981 "Path MTU Discovery for IP version 6" for details about PathMTU.	
Template Description	
If enabled the IPv4/IPv6 processes incoming ICMP "Packet Too Big" messages and stores a MTU value for each destination address.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pathMtuEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00107]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl/TcplpV6MtuConfig
BSW Parameter	BSW Type
TcplpV6PathMtuTimeout	EcucFloatParamDef
BSW Description	
If this value is >0 the IPv6 will reset the MTU value stored for each destination after n seconds. see [RFC1981 5.3. Purging stale PMTU information] Default: 600 seconds (10 minutes)	
Template Description	
If this value is >0 the IPv4/IPv6 will reset the MTU value stored for each destination after n seconds.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pathMtuTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00105]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpCtrl/TcplpVXCtrl/TcplpV6Ctrl
BSW Parameter	BSW Type
TcplpV6NdpConfigRef	EcucReferenceDef
BSW Description	
Reference to Neighbor Discovery Protocol Configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently)	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00102]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig

BSW Parameter		BSW Type	
TcplpDhcpServerConfig		EcucParamConfContainerDef	
BSW Description			
Specifies the configuration parameters of the DHCP Server sub-module.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00187]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpDhcpServerConfig	
BSW Parameter		BSW Type	
TcplpDhcpAddressAssignment		EcucParamConfContainerDef	
BSW Description			
Defines a Ethernet Switch port based IP address assignment.			
Template Description			
Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddressAssignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.dhcpAddressAssignment			
Mapping Rule			Mapping Type
The existence of a dhcpAddressAssignment leads to one container.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00191]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpDhcpServerConfig/TcplpDhcpAddressAssignment	
BSW Parameter		BSW Type	
TcplpDhcpAddressLowerBound		EcucStringParamDef	
BSW Description			
The lower bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned.			
Template Description			
Ipv4DhcpServerConfiguration.addressRangeLowerBound: Lower range of IP addresses to be issued to DHCP clients. IPv4 Address. Notation: 255.255.255.255.			
Ipv6DhcpServerConfiguration.addressRangeLowerBound: Lower range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:....:FFFF.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4DhcpServerConfiguration.addressRangeLowerBound, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6DhcpServerConfiguration.addressRangeLowerBound			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	

valid	[ECUC_Tcplp_00193]
-------	--------------------

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpDhcpServerConfig/TcplpDhcpAddressAssignment	
BSW Parameter	BSW Type	
TcplpDhcpAddressUpperBound	EcucStringParamDef	
BSW Description		
The upper bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned.		
Template Description		
Ipv4DhcpServerConfiguration.addressRangeUpperBound: Upper range of IP addresses to be issued to DHCP clients. P4 Address. Notation: 255.255.255.255.		
Ipv6DhcpServerConfiguration.addressRangeUpperBound: Upper range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:....FFFF.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4DhcpServerConfiguration.addressRangeUpperBound, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6DhcpServerConfiguration.addressRangeUpperBound		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00194]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpDhcpServerConfig/TcplpDhcpAddressAssignment	
BSW Parameter	BSW Type	
TcplpDhcpSwitchPortRef	EcucReferenceDef	
BSW Description		
Reference to Ethernet Switch port. Optional in case the Dhcp server is operating without an Ethernet switch.		
Template Description		
A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::CouplingPort		
Mapping Rule		Mapping Type
The Switch CouplingPort the VlanMembership.dhcpAddressAssignment belongs to.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00192]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpDhcpServerConfig	
BSW Parameter	BSW Type	
TcplpDhcpDefaultRouter	EcucStringParamDef	
BSW Description		
IP address of default router (gateway).		

Template Description	
Ipv4DhcpServerConfiguration.defaultGateway: IP address of the default gateway. Notation 255.255.255.255	
Ipv6DhcpServerConfiguration.defaultGateway: IP address of the default gateway. Notation 255.255.255.255	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4DhcpServerConfiguration.defaultGateway, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6DhcpServerConfiguration.defaultGateway	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00190]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpDhcpServerConfig
BSW Parameter	BSW Type
TcplpDhcpEthIfSwitchRef	EcucReferenceDef
BSW Description	
Reference to EthIfSwitch representation.	
Optional in case the Dhcp server is operating without an Ethernet switch.	
Template Description	
Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddressAssignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.dhcpAddressAssignment	
Mapping Rule	Mapping Type
The Switch CouplingElement the VlanMembership.dhcpAddressAssignment belongs to.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00188]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpDhcpServerConfig
BSW Parameter	BSW Type
TcplpDhcpNetmask	EcucIntegerParamDef
BSW Description	
Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.	
Template Description	
Ipv4DhcpServerConfiguration.networkMask: Default network mask to be used by DHCP clients. Notation 255.255.255.255	
Ipv6DhcpServerConfiguration.networkMask: Default network mask to be used by DHCP clients. Notation 255.255.255.255	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4DhcpServerConfiguration.networkMask, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6DhcpServerConfiguration.networkMask	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00189]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpDuplicateAddressDetectionConfig	EcucParamConfContainerDef
BSW Description	
Specifies the DAD callout function.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00214]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpDuplicateAddressDetectionConfig
BSW Parameter	BSW Type
TcplpDuplicateAddressDetectionCalloutName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the DAD callout function <Up_DADAddressConflict>.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00216]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpIpConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters of the IP (Internet Protocol) sub-module	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00022]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig	
BSW Parameter	BSW Type	
TcplpV4Config	EcucParamConfContainerDef	
BSW Description		
Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00095]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpV4Config	
BSW Parameter	BSW Type	
TcplpArpConfig	EcucParamConfContainerDef	
BSW Description		
Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00023]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpV4Config/TcplpArpConfig	
BSW Parameter	BSW Type	
TcplpArpNumGratuitousARPOnStartup	EcucIntegerParamDef	
BSW Description		
Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00054]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config/TcplpArpConfig	
BSW Parameter	BSW Type	
TcplpArpPacketQueueEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00170]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config/TcplpArpConfig	
BSW Parameter	BSW Type	
TcplpArpRequestTimeout	EcucFloatParamDef	
BSW Description	Specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the Tcplp shall skip the transmission of any further ARP requests to the same destination within a duration of TcplpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1) The value for this parameter shall be an integral multiple of TcplpMainFunctionPeriod or 0. If this parameter set to 0 this features is disabled and no delay between ARP requests is enforced.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00218]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config/TcplpArpConfig	
BSW Parameter	BSW Type	
TcplpArpTableEntryTimeout	EcucFloatParamDef	
BSW Description	Timeout in seconds after which an unused ARP entry is removed.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00053]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config/TcplpArpConfig	
BSW Parameter	BSW Type	
TcplpArpTableSizeMax	EcucIntegerParamDef	
BSW Description	Maximum number of entries in the ARP table.	
Template Description	This attribute specifies the size of neighbor cache or ARP table in units of entries.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.neighborCacheSize	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00052]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config	
BSW Parameter	BSW Type	
TcplpAutolpConfig	EcucParamConfContainerDef	
BSW Description	Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00028]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config/TcplpAutolpConfig	
BSW Parameter	BSW Type	
TcplpAutolpInitTimeout	EcucFloatParamDef	
BSW Description	The time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00074]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV4Config	
BSW Parameter	BSW Type	

TcplpDhcpConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters of the DHCPv4.	
This container may be referenced by multiple IPv4 instances if they shall use the same configuration.	
This container may have multiple instances if different configurations are required for different IPv4 instances.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00167]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config
BSW Parameter	BSW Type
TcplpIcmpConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00024]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpIcmpConfig
BSW Parameter	BSW Type
TcplpIcmpEchoReplyEnabled	EcucBooleanParamDef
BSW Description	
Enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00213]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpIcmpConfig
BSW Parameter	BSW Type

TcplplcmpMsgHandler	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMP message handler.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00056]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV4Config/TcplplcmpConfig/TcplplcmpMsgHandler
BSW Parameter	BSW Type
TcplplcmpMsgHandlerName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the ICMP message handler function <Up_lcmpMsgHandler>.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00057]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV4Config/TcplplcmpConfig
BSW Parameter	BSW Type
TcplplcmpTtl	EcucIntegerParamDef
BSW Description	
Default Time-to-live value of outgoing ICMP packets.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00055]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV4Config
BSW Parameter	BSW Type
TcplplpFragmentationConfig	EcucParamConfContainerDef
BSW Description	

Specifies the configuration parameters of IPv4 packet fragmentation/reassembly.

This container may be referenced by multiple IPv4 instances if they shall use the same configuration.
This container may have multiple instances if different configurations are required for different IPv4 instances.

Template Description

M2 Parameter

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00108]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpFragmentation Config	
BSW Parameter		BSW Type
TcplpFragmentationRxEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00077]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpFragmentation Config	
BSW Parameter		BSW Type
TcplpNumFragments		EcucIntegerParamDef
BSW Description		
Specifies the maximum number of IP fragments per datagram. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00078]	

BSW Module	BSW Context

Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpFragmentation Config	
BSW Parameter		BSW Type
TcplpNumReassDgrams		EcucIntegerParamDef
BSW Description		
Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00080]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV4Config/TcplpFragmentation Config	
BSW Parameter		BSW Type
TcplpReassTimeout		EcucFloatParamDef
BSW Description		
Specifies the timeout in [s] after which an incomplete datagram gets discarded. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00079]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig	
BSW Parameter		BSW Type
TcplpV6Config		EcucParamConfContainerDef
BSW Description		
Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00168]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config	

BSW Parameter		BSW Type	
TcplpDhcpV6Config		EcucParamConfContainerDef	
BSW Description			
Specifies the configuration parameters of the DHCPv6.			
This container may be referenced by multiple IPv6 instances if they shall use the same configuration.			
This container may have multiple instances if different configurations are required for different IPv6 instances.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00110]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config		
BSW Parameter		BSW Type	
TcplpDhcpV6CnfDelayMax		EcucFloatParamDef	
BSW Description			
Maximum delay (s) before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00116]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config		
BSW Parameter		BSW Type	
TcplpDhcpV6CnfDelayMin		EcucFloatParamDef	
BSW Description			
Minimum delay (s) before the first Confirm message will be sent.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00117]	

BSW Module	BSW Context
------------	-------------

Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config	
BSW Parameter		BSW Type
TcplpDhcpV6InfDelayMax		EcucFloatParamDef
BSW Description		
Maximum delay (s) before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00118]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config	
BSW Parameter		BSW Type
TcplpDhcpV6InfDelayMin		EcucFloatParamDef
BSW Description		
Minimum delay (s) before the first Information Request message will be sent.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00119]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config	
BSW Parameter		BSW Type
TcplpDhcpV6SolDelayMax		EcucFloatParamDef
BSW Description		
Maximum delay (s) before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00120]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpDhcpV6Config	
BSW Parameter		BSW Type
TcplpDhcpV6SolDelayMin		EcucFloatParamDef

BSW Description	
Minimum delay (s) before the first Solicit message will be sent.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00121]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config
BSW Parameter	BSW Type
TcplpcmpV6Config	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00113]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpcmpV6Config
BSW Parameter	BSW Type
TcplpcmpV6EchoReplyAvoidFragmentation	EcucBooleanParamDef
BSW Description	
If enabled, the stack will respond only to incoming ICMPv6 Echo Requests (Pings) that fit the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation. Only relevant if TcplpcmpV6EchoReplyEnabled is enabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00212]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpcmpV6Config
BSW Parameter	BSW Type
TcplpcmpV6EchoReplyEnabled	EcucBooleanParamDef
BSW Description	
If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings).	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00149]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV6Config/TcplplcmpV6Config
BSW Parameter	BSW Type
TcplplcmpV6HopLimit	EcucIntegerParamDef
BSW Description	
Default Hop-Limit value of outgoing ICMPv6 packets.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00152]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV6Config/TcplplcmpV6Config
BSW Parameter	BSW Type
TcplplcmpV6MsgDestinationUnreachableEnabled	EcucBooleanParamDef
BSW Description	
Dis/Enables transmission of Destination Unreachable Messages	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00153]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplplpConfig/TcplplpV6Config/TcplplcmpV6Config
BSW Parameter	BSW Type
TcplplcmpV6MsgHandler	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00154]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpIcmpV6Config/TcplpIcmpV6MsgHandler	
BSW Parameter	BSW Type	
TcplpIcmpV6MsgHandlerName	EcucFunctionNameDef	
BSW Description		
This parameter defines the name of the ICMP message handler function <Up_IcmpMsgHandler>.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00156]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpIcmpV6Config	
BSW Parameter	BSW Type	
TcplpIcmpV6MsgParameterProblemEnabled	EcucBooleanParamDef	
BSW Description		
If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet.		
[RFC8200 4. IPv6 Extension Headers]		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00151]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config	
BSW Parameter	BSW Type	
TcplpV6ConfigExtHeaderFilter	EcucParamConfContainerDef	
BSW Description		
This container describes the white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed here shall be silently dropped.		
Template Description		
White list for the filtering of IPv6 extension headers.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::IPv6HeaderFilterList::IPv6ExtHeaderFilterList		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00198]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpV6ConfigExtHeaderFilter	
BSW Parameter	BSW Type	
TcplpV6ConfigExtHeaderFilterEntry	EcucIntegerParamDef	
BSW Description		
IPv6 Extension Header type allowed by this filter.		
Template Description		
IPv6 Extension Header type allowed by this filter.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::IPv6HeaderFilterList::IPv6ExtHeaderFilterList.allowedIPv6ExtHeader		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00199]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config	
BSW Parameter	BSW Type	
TcplpV6FragmentationConfig	EcucParamConfContainerDef	
BSW Description		
Specifies the configuration parameters of IPv6 packet fragmentation/reassembly.		
This container may be referenced by multiple IPv6 instances if they shall use the same configuration.		
This container may have multiple instances if different configurations are required for different IPv6 instances.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00114]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpV6FragmentationConfig	
BSW Parameter	BSW Type	
TcplpV6ReassemblyBufferCount	EcucIntegerParamDef	
BSW Description		

<p>Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded.</p> <p>A value of 0 disables fragment reassembly.</p> <p>[RFC8200 5. Packet Size Issues] "In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets)."</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00157]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpV6Config/TcplpV6Fragmentation Config
BSW Parameter	BSW Type
TcplpV6ReassemblyBufferSize	EcucIntegerParamDef
BSW Description	
<p>[RFC8200 5. Packet Size Issues] "A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. A node is permitted to accept fragmented packets that reassemble to more than 1500 octets."the measured path MTU (i.e., down to 1280 octets)."</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00158]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpV6Config/TcplpV6Fragmentation Config
BSW Parameter	BSW Type
TcplpV6ReassemblySegmentCount	EcucIntegerParamDef
BSW Description	
<p>Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed.</p> <p>To deal with fragments received out of order this value should be configured bigger than 1.</p>	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00160]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpV6Fragmentation Config	
BSW Parameter		BSW Type
TcplpV6ReassemblyTimeout		EcucFloatParamDef
BSW Description		
[RFC8200 4.5 Fragment Header] Default: 60 seconds		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00159]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpV6Fragmentation Config	
BSW Parameter		BSW Type
TcplpV6TxFragmentBufferCount		EcucIntegerParamDef
BSW Description		
These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented.		
A value of 0 disables tx fragmentation.		
If the upper layer transmits packets that do not fit into the link or path MTU, the IpV6 will split-up the packet into fragments.		
see "Enable Fragment Reassembly"		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00161]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpV6Fragmentation Config	
BSW Parameter		BSW Type

TcplpV6TxFragmentBufferSize	EcucIntegerParamDef
BSW Description	
Size of each fragment tx buffer in bytes	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00162]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config
BSW Parameter	BSW Type
TcplpNdpConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6	
This container may be referenced by multiple IPv6 instances if they shall use the same configuration.	
This container may have multiple instances if different configurations are required for different IPv6 instances.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00112]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig
BSW Parameter	BSW Type
TcplpNdpArNudConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00123]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig

BSW Parameter		BSW Type
TcplpNdpDefaultReachableTime		EcucFloatParamDef
BSW Description		
<p>Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].</p> <p>"The time a neighbor is considered reachable after receiving a reachability confirmation."</p> <p>If "TcplpNdpDynamicReachableTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements.</p> <p>Default: REACHABLE_TIME = 30 seconds</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00130]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter		BSW Type
TcplpNdpDefaultRetransTimer		EcucFloatParamDef
BSW Description		
<p>Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].</p> <p>"The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor."</p> <p>If "TcplpNdpDynamicRetransTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements.</p> <p>Default: RETRANS_TIMER = 1 second</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00165]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter		BSW Type
TcplpNdpDefensiveProcessing		EcucBooleanParamDef
BSW Description		

<p>If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861. [RFC4861 7.2.5. Receipt of Neighbor Advertisements]</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00201]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig
BSW Parameter	BSW Type
TcplpNdpDelayFirstProbeTime	EcucFloatParamDef
BSW Description	
<p>Delay before sending the first NUD probe in (s).</p> <p>[RFC4861 7.3.3. Node Behavior]</p> <p>Default: DELAY_FIRST_PROBE_TIME = 5 seconds</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00133]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig
BSW Parameter	BSW Type
TcplpNdpMaxNeighborCacheSize	EcucIntegerParamDef
BSW Description	
<p>Maximum number of entries in the neighbor cache.</p> <p>[RFC4861 5.1. Conceptual Data Structures]</p>	
Template Description	
This attribute specifies the size of neighbor cache or ARP table in units of entries.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.neighborCacheSize	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00129]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter		BSW Type
TcplpNdpMaxRandomFactor		EcucIntegerParamDef
BSW Description		
Maximum random factor used for randomization		
[RFC4861 10. Protocol Constants]		
Default: 15 (MAX_RANDOM_FACTOR = 1.5)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00135]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter		BSW Type
TcplpNdpMinRandomFactor		EcucIntegerParamDef
BSW Description		
Minimum random factor used for randomization		
[RFC4861 10. Protocol Constants]		
Default: 5 (MIN_RANDOM_FACTOR = 0.5)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00134]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter		BSW Type
TcplpNdpNeighborUnreachabilityDetectionEnabled		EcucBooleanParamDef
BSW Description		
Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00136]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter	BSW Type	
TcplpNdpNumMulticastSolicitations	EcucIntegerParamDef	
BSW Description		
Maximum number of multicast solicitations that will be sent when performing address resolution. [RFC4861 7.2.2. Sending Neighbor Solicitations] Default: MAX_MULTICAST_SOLICIT = 3		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00132]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter	BSW Type	
TcplpNdpNumUnicastSolicitations	EcucIntegerParamDef	
BSW Description		
Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection. [RFC4861 7.3.3. Node Behavior] Default: MAX_UNICAST_SOLICIT = 3		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00131]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig	
BSW Parameter	BSW Type	
TcplpNdpPacketQueueEnabled	EcucBooleanParamDef	

BSW Description	
Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00171]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpArNudConfig
BSW Parameter	BSW Type
TcplpNdpRandomReachableTimeEnabled	EcucBooleanParamDef
BSW Description	
If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time	
[RFC4861 6.3.2. Host Variables / ReachableTime]	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00137]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig
BSW Parameter	BSW Type
TcplpNdpPrefixRouterDiscoveryConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters for NDP Prefix and Router Discovery.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00124]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig

BSW Parameter		BSW Type	
TcplpNdpDefaultRouterListSize		EcucIntegerParamDef	
BSW Description			
Maximum number of default router entries. [RFC4861 5.1. Conceptual Data Structures]			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00139]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig		
BSW Parameter		BSW Type	
TcplpNdpDestinationCacheSize		EcucIntegerParamDef	
BSW Description			
Maximum number of entries in the destination cache. [RFC4861 5.1. Conceptual Data Structures]			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00138]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig		
BSW Parameter		BSW Type	
TcplpNdpDynamicHopLimitEnabled		EcucBooleanParamDef	
BSW Description			
If enabled the default hop limit may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements]			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00147]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpDynamicMtuEnabled		EcucBooleanParamDef
BSW Description		
Allow dynamic reconfiguration of link MTU via Router Advertisements.		
[RFC4861 4.6.4. MTU]		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00148]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpDynamicReachableTimeEnabled		EcucBooleanParamDef
BSW Description		
If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.		
[RFC4861 6.3.4. Processing Received Router Advertisements]		
Default: Enabled		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00146]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpDynamicRetransTimeEnabled		EcucBooleanParamDef
BSW Description		
If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.		
[RFC4861 6.3.4. Processing Received Router Advertisements]		
Default: Enabled		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00145]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig
BSW Parameter	BSW Type
TcplpNdpMaxRtrSolicitationDelay	EcucFloatParamDef
BSW Description	
Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).	
[RFC4861 6.3.7. Sending Router Solicitations]	
Default: MAX_RTR_SOLICITATION_DELAY = 1 second	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00143]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig
BSW Parameter	BSW Type
TcplpNdpMaxRtrSolicitations	EcucIntegerParamDef
BSW Description	
Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.	
0 = No Router Solicitations will be sent.	
This has no impact on handling Router Advertisements.	
[RFC4861 6.3.7. Sending Router Solicitations]	
Default: MAX_RTR_SOLICITATIONS = 3 transmissions	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00142]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpPrefixList		EcucParamConfContainerDef
BSW Description		
Specifies a list of prefixes to be treated as "on-link" according to IETF RFC 4861 Section 5.1.		
Template Description		
Internet Protocol version 6 (IPv6) configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration		
Mapping Rule		Mapping Type
<p>A distinct list of all prefixes used within the same local network shall be retrieved for the respective ECU configuration.</p> <p>This can be achieved by following all socket connections of this ECU, identify the communication partners and their NetworkEndPoint elements via the ApplicationEndpoint references, retrieve the prefixes of NetworkEndPoint/Ipv6Configuration/ipAddressPrefixLength and NetworkEndPoint/Ipv6Configuration/ipv6Address and create a distinct list of them.</p>		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00205]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig/TcplpNdpPrefixList	
BSW Parameter		BSW Type
TcplpNdpPrefixListEntry		EcucParamConfContainerDef
BSW Description		
Single entry in the prefix list.		
Template Description		
Internet Protocol version 6 (IPv6) configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration		
Mapping Rule		Mapping Type
see upstream mapping in TcplpNdpPrefixList		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00206]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig/TcplpNdpPrefixList/TcplpNdpPrefixListEntry	
BSW Parameter		BSW Type
TcplpNdpPrefixListEntryPrefixAddress		EcucStringParamDef
BSW Description		
The prefix of an IP address. This prefix can be used for on-link determination.		
Template Description		
Internet Protocol version 6 (IPv6) configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration		
Mapping Rule		Mapping Type
see upstream mapping in TcplpNdpPrefixList		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00208]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig/TcplpNdpPrefixList/TcplpNdpPrefixListEntry	
BSW Parameter		BSW Type
TcplpNdpPrefixListEntryPrefixLength		EcucIntegerParamDef
BSW Description		
The number of leading bits in the Prefix that are valid.		
Template Description		
Internet Protocol version 6 (IPv6) configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration		
Mapping Rule		Mapping Type
see upstream mapping in TcplpNdpPrefixList		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00207]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpPrefixListSize		EcucIntegerParamDef
BSW Description		
Maximum number of entries in the on-link prefix list.		
[RFC4861 5.1. Conceptual Data Structures]		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00140]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpConfig/TcplpIpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig	
BSW Parameter		BSW Type
TcplpNdpRndRtrSolicitationDelayEnabled		EcucBooleanParamDef
BSW Description		
If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds.		
[RFC4861 6.3.7. Sending Router Solicitations]		
Default: Enabled		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00141]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpPrefixRouterDiscoveryConfig
BSW Parameter	BSW Type
TcplpNdpRtrSolicitationInterval	EcucFloatParamDef
BSW Description	
Interval between consecutive Router Solicitations in (s).	
[RFC4861 6.3.7. Sending Router Solicitations]	
Default: RTR_SOLICITATION_INTERVAL = 4 seconds	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00144]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig
BSW Parameter	BSW Type
TcplpNdpSlaacConfig	EcucParamConfContainerDef
BSW Description	
Specifies the configuration parameters for StateLess Address AutoConfiguration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00122]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpSlaacConfig
BSW Parameter	BSW Type
TcplpNdpSlaacDadNumberOfTransmissions	EcucIntegerParamDef
BSW Description	

Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigured address to PREFERRED (usable) state.	
[RFC4861 5.1. Node Configuration Variables]	
Default: DupAddrDetectTransmits = 1	
Setting this value to 0 turns off DAD.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00128]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpSlaacConfig
BSW Parameter	BSW Type
TcplpNdpSlaacDadRetransmissionDelay	EcucFloatParamDef
BSW Description	
Sets the maximum value for the address configuration delay (s).	
According to [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] this value should be the same as MAX_RTR_SOLICITATION_DELAY.	
Default: MAX_RTR_SOLICITATION_DELAY = 1 second	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00127]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpSlaacConfig
BSW Parameter	BSW Type
TcplpNdpSlaacDelayEnabled	EcucBooleanParamDef
BSW Description	

If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].

"This serves to alleviate congestion when many nodes start up on the link at the same time, such as after a power failure, and may help to avoid race conditions when more than one node is trying to solicit for the same address at the same time."

"The delay will avoid similar congestion when multiple nodes are going to configure addresses by receiving the same single multicast router advertisement."

[RFC4861 5.4.2. Sending Neighbor Solicitation Messages]

Default: True

Template Description

M2 Parameter

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00125]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpConfig/TcplpV6Config/TcplpNdpConfig/TcplpNdpSlaacConfig	
BSW Parameter		BSW Type
TcplpNdpSlaacOptimisticDadEnabled		EcucBooleanParamDef
BSW Description		
Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00126]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig	
BSW Parameter		BSW Type
TcplpIpSecConfigSet		EcucParamConfContainerDef
BSW Description		
Specifies the IPsec configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00288]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet	
BSW Parameter		BSW Type
TcplpEncryptionAlgorithm		EcucParamConfContainerDef
BSW Description		
Container for configuration of supported encryption algorithm transforms. This container is used to configure supported algorithms for ESP. The transform algorithm must be configured in the Crypto module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00317]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpEncryptionAlgorithm	
BSW Parameter		BSW Type
TcplpEncryptionTransformIdentifier		EcucEnumerationParamDef
BSW Description		
Encryption algorithm transform identifier. Parameter values are defined as per IETF RFC 7296 3.3.2		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00311]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpEncryptionAlgorithm	
BSW Parameter		BSW Type
TcplpEncryptionTransformJobPair		EcucParamConfContainerDef
BSW Description		
Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00312]

BSW Module	BSW Context

Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpEncryptionAlgorithm/TcplpEncryptionTransformJobPair	
BSW Parameter		BSW Type
TcplpCsmDecryptJobRef		EcucReferenceDef
BSW Description		
The referenced Csm job is used for the execution of the CsmMacVerify primitive needed for this transform. Must be a valid decryption job of the parent type.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00314]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpEncryptionAlgorithm/TcplpEncryptionTransformJobPair	
BSW Parameter		BSW Type
TcplpCsmEncryptJobRef		EcucReferenceDef
BSW Description		
The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform. Must be a valid encryption job of the parent type.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00313]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet	
BSW Parameter		BSW Type
TcplpIntegrityAlgorithm		EcucParamConfContainerDef
BSW Description		
Container for configuration of supported integrity algorithm transforms. This container is used to configure supported algorithms for AH. The transform algorithm must be configured in the Crypto module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00294]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpIntegrityAlgorithm	
BSW Parameter		BSW Type
TcplpIntegrityTransformIdentifier		EcucEnumerationParamDef
BSW Description		
Integrity algorithm transform identifier. Parameter values are defined as per IETF RFC 7296 3.3.2		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00307]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpIntegrityAlgorithm	
BSW Parameter		BSW Type
TcplpIntegrityTransformJobPair		EcucParamConfContainerDef
BSW Description		
Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00308]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpIntegrityAlgorithm/TcplpIntegrityTransformJobPair	
BSW Parameter		BSW Type
TcplpCsmGenerateJobRef		EcucReferenceDef
BSW Description		
The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform. Must be a valid MAC generate job of the parent type.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00309]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpIntegrityAlgorithm/TcplpIntegrityTransformJobPair	
BSW Parameter		BSW Type
TcplpCsmVerifyJobRef		EcucReferenceDef
BSW Description		
The referenced Csm job is used for the execution of the CsmMacVerify primitive needed for this transform. Must be a valid MAC verify job of the parent type.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00310]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet	
BSW Parameter		BSW Type
TcplpSecAuditEventCalloutFunction		EcucFunctionNameDef
BSW Description		
This parameter specifies the name of a callout function that will be called for each auditable event.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00292]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet	
BSW Parameter		BSW Type
TcplpSecCalloutHeaderFile		EcucStringParamDef
BSW Description		
This parameter specifies the name of the header file containing the definition for the functions specified in TcplpSecSpdCalloutFunction and TcplpSecAuditEvent		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00291]

BSW Module	BSW Context

Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet	
BSW Parameter		BSW Type
TcplpIpSecSpdCalloutFunction		EcucFunctionNameDef
BSW Description		
This parameter specifies the name of a callout function that shall be called for each Rx/Tx message, after the IPsec has processed all corresponding SPD entries and has determined the policy. The callout function allows it to override the applied policy.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00290]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet	
BSW Parameter		BSW Type
TcplpRemoteDeviceNum		EcucIntegerParamDef
BSW Description		
Amount of remote clients which will negotiate a Security Association (SA).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00289]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet	
BSW Parameter		BSW Type
TcplpSpdEntry		EcucParamConfContainerDef
BSW Description		
Entry of the Security Policy Database (SPD).		
Template Description		
This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.		
M2 Parameter		
SystemTemplate::SecureCommunication::IPSecRule		
Mapping Rule		Mapping Type
Each SpdEntry shall be derived from the IPSecRules that are defined on a NetworkEndpoint that defines the local IP Address range.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00293]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry	

BSW Parameter		BSW Type	
TcplpIpSecHeaderType		EcucEnumerationParamDef	
BSW Description			
Header type specifying the IPsec security mechanism.			
Template Description			
Header type specifying the IPsec security mechanism.			
M2 Parameter			
SystemTemplate::SecureCommunication::IPSecRule.headerType			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00297]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry/TcplpIpSecHeaderType	
BSW Parameter		BSW Type	
TCPIP_IPSEC_HDR_AH		EcucEnumerationLiteralDef	
BSW Description			
Template Description			
Authentication Header (AH)			
M2 Parameter			
SystemTemplate::SecureCommunication::IPsecHeaderTypeEnum.ah			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry/TcplpIpSecHeaderType	
BSW Parameter		BSW Type	
TCPIP_IPSEC_HDR_ESP		EcucEnumerationLiteralDef	
BSW Description			
Template Description			
Encapsulating Security Payloads (ESP)			
M2 Parameter			
SystemTemplate::SecureCommunication::IPsecHeaderTypeEnum.esp			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry/TcplpIpSecHeaderType	
BSW Parameter		BSW Type	
TCPIP_IPSEC_HDR_NONE		EcucEnumerationLiteralDef	

BSW Description	
Template Description	
No header	
M2 Parameter	
SystemTemplate::SecureCommunication::IPsecHeaderTypeEnum.none	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpSecPolicy	EcucEnumerationParamDef
BSW Description	
Policy for usage of IPsec.	
Template Description	
An IPsec policy defines the rules that determine which type of IP traffic needs to be secured using IPsec and how that traffic is secured.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.policy	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00295]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpSecPolicy
BSW Parameter	BSW Type
TCPIP_IPSEC_POLICY_BYPASS	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Signifying that no IPsec processing should be done at all.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPsecPolicyEnum.passthrough	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpSecPolicy
BSW Parameter	BSW Type
TCPIP_IPSEC_POLICY_DISCARD	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Signifying that packets should be discarded	

M2 Parameter	
SystemTemplate::SecureCommunication::IPsecPolicyEnum.drop	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpSecPolicy
BSW Parameter	BSW Type
TCPIP_IPSEC_POLICY_OPTIONAL	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Signifying that packets should be discarded and a diagnostic ICMP returned.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPsecPolicyEnum.reject	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpSecPolicy
BSW Parameter	BSW Type
TCPIP_IPSEC_POLICY_PROTECT	EcucEnumerationLiteralDef
BSW Description	
Template Description	
Signifying that packets should be protected.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPsecPolicyEnum.ipsec	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpSecPriority	EcucIntegerParamDef
BSW Description	
Priority of the SPD entry. The processing of entries is based on priority, starting with the highest priority "0". The first matching SPD entry defines the policy.	
Template Description	
This attribute defines the priority of the IPsecRule (SPD entry). The processing of entries is based on priority, starting with the highest priority "0".	
M2 Parameter	
SystemTemplate::SecureCommunication::IPsecRule.priority	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00296]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry	
BSW Parameter	BSW Type	
TcplpLocalIpAddrEnd	EcucStringParamDef	
BSW Description		
End value of the remote IP address range.		
Template Description		
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpoint		
Mapping Rule		Mapping Type
The local IP address range shall be derived from the NetworkEndpoint that aggregates the IPsecConfig and the collection of IPsecRules from which the Spd entries are derived. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00301]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry	
BSW Parameter	BSW Type	
TcplpLocalIpAddrStart	EcucStringParamDef	
BSW Description		
Start value of the local IP address range.		
Template Description		
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpoint		
Mapping Rule		Mapping Type
The local IP address range shall be derived from the NetworkEndpoint that aggregates the IPsecConfig and the collection of IPsecRules from which the Spd entries are derived. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00300]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry	
BSW Parameter	BSW Type	
TcplpLocalPortRangeEnd	EcucIntegerParamDef	
BSW Description		
End value of the local port range.		
Template Description		

This attribute restricts the traffic monitoring to tcp and defines a end value for the local port range.	
If this attribute is not set then this rule shall be effective for all local ports.	
Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.localPortRangeEnd	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00299]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpLocalPortRangeStart	EcucIntegerParamDef
BSW Description	
Start value of the local port range.	
Template Description	
This attribute restricts the traffic monitoring to tcp and defines a start value for the local port range.	
If this attribute is not set then this rule shall be effective for all local ports.	
Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.localPortRangeStart	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00298]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpIpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpProtocol	EcucEnumerationParamDef
BSW Description	
Relevant IP protocol. Note: As specified in IETF Rfc 4301 section 6, ICMP error messages will always be BYPASSEd. The policy for TCPIP_PROTOCOL_ICMP only applies to ICMP non-error messages. (Echo reply/re-sponse).	
Template Description	
This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.ipProtocol	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00306]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpProtocol	
BSW Parameter	BSW Type	
TCPIP_PROTOCOL_ANY	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
ANY protocol		
M2 Parameter		
SystemTemplate::SecureCommunication::IPseclpProtocolEnum.any		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpProtocol	
BSW Parameter	BSW Type	
TCPIP_PROTOCOL_ICMP	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
Internet Control Message Protocol (ICMP)		
M2 Parameter		
SystemTemplate::SecureCommunication::IPseclpProtocolEnum.icmp		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpProtocol	
BSW Parameter	BSW Type	
TCPIP_PROTOCOL_TCP	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
TCP Protocol		
M2 Parameter		
SystemTemplate::SecureCommunication::IPseclpProtocolEnum.tcp		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry/TcplpProtocol	
BSW Parameter	BSW Type	
TCPIP_PROTOCOL_UDP	EcucEnumerationLiteralDef	
BSW Description		

Template Description	
UDP Protocol	
M2 Parameter	
SystemTemplate::SecureCommunication::IPseclpProtocolEnum.udp	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpRemIpAddrEnd	EcucStringParamDef
BSW Description	
End value of the remote IP address range.	
Template Description	
Definition of the remote NetworkEndpoint. With this reference the connection between the local NetworkEndpoint and the remote NetworkEndpoint is described on which the traffic is monitored.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.remotelpAddress	
Mapping Rule	Mapping Type
The remote IP address range shall be derived from the NetworkEndpoints that are referenced by IPSecRule.remotelpAddress. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00303]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry
BSW Parameter	BSW Type
TcplpRemIpAddrStart	EcucStringParamDef
BSW Description	
Start value of the remote IP address range.	
Template Description	
Definition of the remote NetworkEndpoint. With this reference the connection between the local NetworkEndpoint and the remote NetworkEndpoint is described on which the traffic is monitored.	
M2 Parameter	
SystemTemplate::SecureCommunication::IPSecRule.remotelpAddress	
Mapping Rule	Mapping Type
The remote IP address range shall be derived from the NetworkEndpoints that are referenced by IPSecRule.remotelpAddress. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00302]

BSW Module	BSW Context
-------------------	--------------------

Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry	
BSW Parameter		BSW Type
TcplpRemPortRangeEnd	EcucIntegerParamDef	
BSW Description		
End value of the remote port range.		
Template Description		
This attribute restricts the traffic monitoring to tcp and defines a end value for the remote port range.		
If this attribute is not set then this rule shall be effective for all local ports.		
Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.		
M2 Parameter		
SystemTemplate::SecureCommunication::IPSecRule.remotePortRangeEnd		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00305]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSecConfigSet/TcplpSpdEntry	
BSW Parameter		BSW Type
TcplpRemPortRangeStart	EcucIntegerParamDef	
BSW Description		
Start value of the remote port range.		
Template Description		
This attribute restricts the traffic monitoring to tcp and defines a start value for the remote port range.		
If this attribute is not set then this rule shall be effective for all local ports.		
Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.		
M2 Parameter		
SystemTemplate::SecureCommunication::IPSecRule.remotePortRangeStart		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00304]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig	
BSW Parameter		BSW Type
TcplpLocalAddr	EcucParamConfContainerDef	
BSW Description		
Specifies the local IP (Internet Protocol) addresses used for IP communication.		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress		
Mapping Rule		Mapping Type

Create container for each NetworkEndpointAddress element that is defined in the ECU Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00020]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	
BSW Parameter	BSW Type	
TcplpAddrAssignment	EcucParamConfContainerDef	
BSW Description		
This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00033]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment	
BSW Parameter	BSW Type	
TcplpAssignmentLifetime	EcucEnumerationParamDef	
BSW Description		
Defines the lifetime of a dynamically fetched IP address.		
If TcplpAssignmentMethod = TCPIP_STATIC then TcplpAssignmentLifetime shall be omitted.		
Template Description		
Ipv4Configuration.ipAddressKeepBehavior: Defines the lifetime of a dynamically fetched IP address.		
Ipv6Configuration.ipAddressKeepBehavior: Defines the lifetime of a dynamically fetched IP address.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.ipAddressKeepBehavior, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipAddressKeepBehavior		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00186]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignmentLifetime	
BSW Parameter	BSW Type	
TCPIP_FORGET	EcucEnumerationLiteralDef	
BSW Description		
After a dynamic IP address has been assigned just use it for this link-up time.		

Template Description	
After a dynamic IP address has been assigned just use it for this session.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::IpAddressKeepEnum.forget	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignmentLifetime
BSW Parameter	BSW Type
TCPIP_STORE	EcucEnumerationLiteralDef
BSW Description	
After a dynamic IP address has been assigned store the address persistently.	
Template Description	
After a dynamic IP address has been assigned store the address persistently.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::IpAddressKeepEnum.storePersistently	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment
BSW Parameter	BSW Type
TcplpAssignmentMethod	EcucEnumerationParamDef
BSW Description	
Method of address assignment	
Template Description	
Ipv4Configuration.ipv4AddressSource: Defines how the node obtains its IP address.	
Ipv6Configuration.ipv6AddressSource: Defines how the node obtains its IP address.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.ipv4AddressSource, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipv6AddressSource	
Mapping Rule	Mapping Type
Derive parameter from the AddressSource attributes.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00035]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignmentMethod
BSW Parameter	BSW Type
TCPIP_DHCP	EcucEnumerationLiteralDef

BSW Description	
Dynamic Assigned IP Address using DHCP	
Template Description	
Ipv4AddressSourceEnum.dhcpv4: DHCP is a service for the automatic IP configuration of a client.	
Ipv6AddressSourceEnum.dhcpv6: DHCP is a service for the automatic IP configuration of a client.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4AddressSourceEnum.dhcpv4, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6AddressSourceEnum.dhcpv6	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignment Method
BSW Parameter	BSW Type
TCPIP_IPV6_ROUTER	EcucEnumerationLiteralDef
BSW Description	
Dynamic Configured IPv6 Address by Router Advertisement	
Template Description	
IPv6 Stateless Autoconfiguration.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6AddressSourceEnum.routerAdver- tisement	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignment Method
BSW Parameter	BSW Type
TCPIP_LINKLOCAL	EcucEnumerationLiteralDef
BSW Description	
Linklocal IPv4/IPv6 Address Assignment	
Template Description	
Ipv4AddressSourceEnum.autolp: AutoIP is used to dynamically assign IP addresses at device startup.	
Ipv6AddressSourceEnum.linkLocal: LinkLocal is intended only for communications within the segment of a local network (a link) or a point-to-point connection that a host is connected to.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4AddressSourceEnum.autolp, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6AddressSourceEnum.linkLocal	
Mapping Rule	Mapping Type
1:1 mapping	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignment Method	
BSW Parameter		BSW Type
TCPIP_LINKLOCAL_DOIP		EcucEnumerationLiteralDef
BSW Description		
Linklocal IPv4/IPv6 Address Assignment using DoIP Parameters		
Template Description		
Ipv4AddressSourceEnum.autolp_doip: Linklocal IPv4 Address Assignment using DoIP Parameters		
Ipv6AddressSourceEnum.linkLocal_doip: Linklocal IPv6 Address Assignment using DoIP Parameters		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4AddressSourceEnum.autolp_doip, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6AddressSourceEnum.linkLocal_doip		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment/TcplpAssignment Method	
BSW Parameter		BSW Type
TCPIP_STATIC		EcucEnumerationLiteralDef
BSW Description		
Static Assigned IP Address		
Template Description		
Ipv4AddressSourceEnum.fixed: The IP Address shall be declared manually.		
Ipv6AddressSourceEnum.fixed: The IP Address shall be declared manually.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4AddressSourceEnum.fixed, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6AddressSourceEnum.fixed		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment	
BSW Parameter		BSW Type
TcplpAssignmentPriority		EcucIntegerParamDef
BSW Description		

Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.	
Template Description	
Ipv4Configuration.assignmentPriority: Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.	
Ipv6Configuration.assignmentPriority: Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.assignmentPriority, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.assignmentPriority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00037]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddrAssignment
BSW Parameter	BSW Type
TcplpAssignmentTrigger	EcucEnumerationParamDef
BSW Description	
Trigger of address assignment.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00036]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr
BSW Parameter	BSW Type
TcplpAddrId	EcucIntegerParamDef
BSW Description	
IP address table identifier assigned by TCP/IP stack.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00029]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	
BSW Parameter	BSW Type	
TcplpAddressType	EcucEnumerationParamDef	
BSW Description		
Address type.		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress		
Mapping Rule	Mapping Type	
shall be derived from the IP Address (see more details in upstream mapping of enum literals).	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00031]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddressType	
BSW Parameter	BSW Type	
TCPIP_ANYCAST	EcucEnumerationLiteralDef	
BSW Description		
Anycast address		
Template Description		
This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.enableAnycast		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddressType	
BSW Parameter	BSW Type	
TCPIP_MULTICAST	EcucEnumerationLiteralDef	
BSW Description		
Multicast address.		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress		
Mapping Rule	Mapping Type	
Shall be set if Multicast Address is used. IPv4: 224.0.0.0 to 239.255.255.255 IPv6: address with the prefix ff00::/8.	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpAddressType	
BSW Parameter	BSW Type	
TCPIP_UNICAST	EcucEnumerationLiteralDef	
BSW Description	Unicast address	
Template Description	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress	
Mapping Rule	Mapping Type	full
Shall be set if Unicast Address is used.		
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	
BSW Parameter	BSW Type	
TcplpCtrlRef	EcucReferenceDef	
BSW Description	Reference to a TcplpCtrl specifying the Ethlf Controller where the IP address shall be assigned.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00032]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	
BSW Parameter	BSW Type	
TcplpDomainType	EcucEnumerationParamDef	
BSW Description	Address family.	
Template Description	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress	
Mapping Rule	Mapping Type	full
Derive this parameter from the NetworkEndpointAddress.		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00030]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpDomainType	
BSW Parameter	BSW Type	

TCPIP_AF_INET	EcucEnumerationLiteralDef
BSW Description	
IPv4 address	
Template Description	
Internet Protocol version 4 (IPv4) configuration.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration	
Mapping Rule	Mapping Type
Set literal to TCPIP_AF_INET when the NetworkEndpoint contains an Ipv4Configuration.	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpDomainType	
BSW Parameter	BSW Type	
TCPIP_AF_INET6	EcucEnumerationLiteralDef	
BSW Description		
IPv6 address		
Template Description		
Internet Protocol version 6 (IPv6) configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration		
Mapping Rule	Mapping Type	
Set literal to TCPIP_AF_INET6 when the NetworkEndpoint contains an Ipv6Configuration.	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	
BSW Parameter	BSW Type	
TcplpLocalAddrIpv6ExtHeaderFilterRef	EcucReferenceDef	
BSW Description		
Reference to a set of IPv6 Extension Headers which are allowed for this local IPv6 address. Note: this parameter is only relevant if the related TcplpDomainType is TCPIP_AF_INET6.		
Template Description		
Reference to a list of IPv6 Extension Headers allowed for this SocketConnection. If no list is referenced all IPv6 Extension Headers are allowed and processed.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SocketAddress.allowedIpv6ExtHeaders		
Mapping Rule	Mapping Type	
1:1 mapping. constraint: All related SocketConnections shall reference either no or exactly the same Ipv6ExtHeaderFilterList.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00200]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr	

BSW Parameter		BSW Type	
TcplpStaticIpAddressConfig		EcucParamConfContainerDef	
BSW Description			
This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00034]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpStaticIpAddressConfig		
BSW Parameter		BSW Type	
TcplpDefaultRouter		EcucStringParamDef	
BSW Description			
IP address of default router (gateway)			
Template Description			
Ipv6Configuration.defaultRouter: IP address of the default router.			
Ipv4Configuration.defaultGateway: IP address of the default gateway.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.defaultRouter, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.defaultGateway			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00040]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpStaticIpAddressConfig		
BSW Parameter		BSW Type	
TcplpNetmask		EcucIntegerParamDef	
BSW Description			
Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.			
Template Description			
Ipv4Configuration.networkMask: Network mask. Notation 255.255.255.255			
Ipv6Configuration.ipAddressPrefixLength: IPv6 prefix length defines the part of the IPv6 address that is the network prefix.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.networkMask, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipAddressPrefixLength			
Mapping Rule		Mapping Type	

- 1:1 mapping for Ipv6 - conversion to CIDR notation for Ipv4	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00039]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpLocalAddr/TcplpStaticIpAddressConfig
BSW Parameter	BSW Type
TcplpStaticIpAddress	EcucStringParamDef
BSW Description	
Static IP Address. To specify any IP address for a certain EthIfCtrl, "ANY" has to be set as wildcard. See Tcplp_Bind() for more details.	
Template Description	
Ipv4Configuration.ipv4Address: IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used.	
Ipv6Configuration.ipv6Address: IPv6 Address. Notation: FFFF::...:FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.ipv4Address, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipv6Address	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00038]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpNvmBlock	EcucParamConfContainerDef
BSW Description	
Configuration of optional usage of Nvm in case the Tcplp module requires non volatile memory in the Ecu to store information (e.g. IP Address received via DHCP and shall be stored).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00184]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpNvmBlock
BSW Parameter	BSW Type
TcplpNvmBlockDescriptorRef	EcucReferenceDef

BSW Description	
Reference to the Nvm block description in the Nvm module configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00185]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpPhysAddrConfig	EcucParamConfContainerDef
BSW Description	
Specifies the physical address configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00083]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpPhysAddrConfig
BSW Parameter	BSW Type
TcplpPhysAddrChgHandler	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of TcplpPhysAddrConfig and specifies the configuration parameters for physical address change handler.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00084]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpPhysAddrConfig/TcplpPhysAddrChgHandler
BSW Parameter	BSW Type
TcplpPhysAddrChgHandlerName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the physical address change function <Up>_PhysAddrTableChg.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00086]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpSocketOwnerConfig	EcucParamConfContainerDef
BSW Description	
Specifies the upper layer modules of Tcplp using the socket API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00172]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig
BSW Parameter	BSW Type
TcplpSocketOwner	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00173]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner
BSW Parameter	BSW Type
TcplpSocketOwnerCopyTxDataName	EcucStringParamDef
BSW Description	
This parameter defines the name of the <Up_CopyTxData> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00180]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerHeaderFileName		EcucStringParamDef
BSW Description		
This parameter specifies the name of the header file containing the definition of the TcplpSocketOwner module functions. The header file name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00175]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerId		EcucIntegerParamDef
BSW Description		
This value specifies the ID of the socket user.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00316]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerLocalIpAddrAssignmentChgName		EcucStringParamDef
BSW Description		
This parameter defines the name of the <Up_LocalIpAddrAssignmentChg> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00181]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter	BSW Type	
TcplpSocketOwnerRxIndicationName	EcucStringParamDef	
BSW Description		
This parameter defines the name of the <Up_RxIndication> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00176]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter	BSW Type	
TcplpSocketOwnerTcpAcceptedName	EcucStringParamDef	
BSW Description		
This parameter defines the name of the <Up_TcpAccepted> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00178]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter	BSW Type	
TcplpSocketOwnerTcpConnectedName	EcucStringParamDef	
BSW Description		
This parameter defines the name of the <Up_TcpConnected> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00179]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerTcplpEventName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <Up_TcplpEvent> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00197]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerTxConfirmationName		EcucStringParamDef
BSW Description		
This parameter defines the name of the <Up_TxConfirmation> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00177]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpSocketOwnerConfig/TcplpSocketOwner	
BSW Parameter		BSW Type
TcplpSocketOwnerUpperLayerType		EcucEnumerationParamDef
BSW Description		
This parameter specifies the type of the upper layer module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00174]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig	
BSW Parameter	BSW Type	
TcplpTcplConfig	EcucParamConfContainerDef	
BSW Description		
Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.		
Template Description		
Content Model for TCP configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp		
Mapping Rule		Mapping Type
This container shall be created if the TcpTp element is used in the ECU Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00025]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcplConfig	
BSW Parameter	BSW Type	
TcplpTcplConfigOptionFilter	EcucParamConfContainerDef	
BSW Description		
This container describes the white list for the filtering of TCP options, i.e. segments containing TCP options not listed here shall be silently dropped.		
Template Description		
White list for the filtering of TCP options.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::TcpOptionFilterSet::TcpOptionFilterList		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00202]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcplConfig/TcplpTcplConfigOptionFilter	
BSW Parameter	BSW Type	
TcplpTcplConfigOptionFilterEntry	EcucIntegerParamDef	
BSW Description		
TCP option kind allowed by this filter.		
Template Description		
TCP option kind allowed by this filter.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::TcpOptionFilterSet::TcpOptionFilterList.allowedTcpOption		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00204]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig/TcplpTcpConfigOptionFilter	
BSW Parameter	BSW Type	
TcplpTcpConfigOptionFilterId	EcucIntegerParamDef	
BSW Description	Identification of the TCP option filter.	
Template Description	TCP option kind allowed by this filter.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::TcpOptionFilterSet::TcpOptionFilterList.allowedTcpOption	
Mapping Rule	Mapping Type	
The list of allowedTcpOptions in TcpOptionFilterList is ordered. The Id of an entry can be derived from the order.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00203]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter	BSW Type	
TcplpTcpCongestionAvoidanceEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00061]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter	BSW Type	
TcplpTcpFastRecoveryEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00063]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter	BSW Type	

TcplpTcpFastRetransmitEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00062]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpFinWait2Timeout	EcucFloatParamDef
BSW Description	
Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00066]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpKeepAliveEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6	
Template Description	
Indicates if Keep-Alive messages are send.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAlives	
Mapping Rule	
	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00082]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpKeepAliveInterval	EcucFloatParamDef

BSW Description	
Specifies the interval in [s] between subsequent keepalive probes.	
Template Description	
Specifies the interval in seconds between subsequent keepalive probes.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAliveInterval	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00070]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpKeepAliveProbesMax	EcucIntegerParamDef
BSW Description	
Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.	
Template Description	
Maximum number of times that TCP retransmits an individual data segment before aborting the connection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAliveProbesMax	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00071]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpKeepAliveTime	EcucFloatParamDef
BSW Description	
Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe. Note: Setting this configuration parameter to a value smaller or equal to the value of TcplpMainFunctionPeriod results in the transmission of keep alive probes within every MainFunction cycle.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00087]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpMaxRtx	EcucIntegerParamDef
BSW Description	

Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if TcplpTcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00069]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpMsl	EcucFloatParamDef
BSW Description	
Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcplpTcpMsl - to ensure that the remote node received the acknowledgment to its connection termination request.)	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00067]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpNagleEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated via Tcplp_ChangeParameter() API.	
Template Description	
Indicates if Nagle's Algorithm is used.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.naglesAlgorithm	
Mapping Rule	Mapping Type
If at least one of the TcpTp.naglesAlgorithm per ApplicationEndpoint is set to true, the parameter shall be set to true. If all of the TcpTp.naglesAlgorithm per ApplicationEndpoint are set to false, the parameter shall be set to false.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00059]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig

BSW Parameter		BSW Type
TcplpTcpReceiveWindowMax		EcucIntegerParamDef
BSW Description		
Default value of maximum receive window in bytes.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00073]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter		BSW Type
TcplpTcpRetransmissionTimeout		EcucFloatParamDef
BSW Description		
Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled or set to INF, no TCP segments shall be retransmitted.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00068]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter		BSW Type
TcplpTcpSlowStartEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00060]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig	
BSW Parameter		BSW Type
TcplpTcpSynMaxRtx		EcucIntegerParamDef
BSW Description		

Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcplpTcpRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcplpTcpRetransmissionTimeout has been expired.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00064]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpSynReceivedTimeout	EcucFloatParamDef
BSW Description	
Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgment after having both received and sent a connection request.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00065]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTcpConfig
BSW Parameter	BSW Type
TcplpTcpTtl	EcucIntegerParamDef
BSW Description	
Default Time-to-live value of outgoing TCP packets.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00072]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig
BSW Parameter	BSW Type
TcplpTlsConfig	EcucParamConfContainerDef
BSW Description	

Specifies the configuration parameters of the TLS (Transport Layer Security) sub module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00219]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig
BSW Parameter	BSW Type
TcplpTlsCiphersuites	EcucParamConfContainerDef
BSW Description	
This container provides the information about supported ciphersuites used by TLS.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00222]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites
BSW Parameter	BSW Type
TcplpTlsCertificateIdentity	EcucParamConfContainerDef
BSW Description	
This container provides information about the certificates used for ciphersuites.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00240]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCertificateIdentity
BSW Parameter	BSW Type
TcplpTlsCipherKeyMLocalCertificate	EcucReferenceDef
BSW Description	
Reference to a KeyM certificate used to address the local certificate.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00286]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCertificate Identity	
BSW Parameter		BSW Type
TcplpTlsCipherKeyMRemoteCertificate		EcucReferenceDef
BSW Description		
Reference to KeyM certificate container to reference the remote certificate.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00287]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCertificate Identity	
BSW Parameter		BSW Type
TcplpTlsServerNameIdentification		EcucStringParamDef
BSW Description		
Defines a server identification name. If present, the name will be added as an extension with the "TLS client hello" handshake message. The TLS server will check for the name to identify the server certificate.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00278]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites	
BSW Parameter		BSW Type
TcplpTlsCiphersuiteDefinition		EcucParamConfContainerDef
BSW Description		
This container provides the static information of a ciphersuite used by TLS.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00237]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsCiphersuiteId		EcucIntegerParamDef
BSW Description		
ID that represents the ciphersuite according to IETF, e.g. RFC4492, Sect. 6, RFC8446, Appendix B.4 or RFC5246, Appendix A.5.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00242]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsCiphersuiteName		EcucStringParamDef
BSW Description		
Provides a verbal name for the ciphersuite. The name should be the one defined in the respective RFC, e.g. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (TLS 1.2) or TLS_AES_128_GCM_SHA256 (TLS 1.3)		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00244]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsCiphersuitePriority		EcucIntegerParamDef
BSW Description		
Defines the priority of the cipher. The higher the number the lower the priority.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00243]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsUseAEADCipher		EcucBooleanParamDef
BSW Description		
Specifies if the ciphersuite supports AEAD for data en-/decryption.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00247]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsUsePresharedKeys		EcucBooleanParamDef
BSW Description		
Defines if this ciphersuite uses pre-shared keys. If so, additional configuration or callbacks will be used for pre-shared key negotiation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00245]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter		BSW Type
TcplpTlsUseSecurityExtensionForceEncryptThenMac		EcucBooleanParamDef
BSW Description		
Defines if the security extension according to IETF RFC 7366 shall be supported. This is useful for ciphersuites using CBC mode.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00246]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Definition	
BSW Parameter	BSW Type	
TcplpTlsVersion	EcucEnumerationParamDef	
BSW Description		
Declares the TLS version that this ciphersuite shall be used for.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00248]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites	
BSW Parameter	BSW Type	
TcplpTlsCiphersuiteWorker	EcucParamConfContainerDef	
BSW Description		
This container provides the jobs and keys necessary for TLS data transmission and reception.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00238]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter	BSW Type	
TcplpTlsCipherAEADCipherKeyLength	EcucIntegerParamDef	
BSW Description		
Defines the key length for en- / decryption with authentication data (AEAD).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00254]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter	BSW Type	
TcplpTlsCipherCsmDecryptJobRef	EcucReferenceDef	
BSW Description	Reference to a CSM job to perform the data decryption operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00255]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter	BSW Type	
TcplpTlsCipherCsmDecryptKeyRef	EcucReferenceDef	
BSW Description	Reference to a CSM key associated to the CSM job that performs the data decryption operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00256]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter	BSW Type	
TcplpTlsCipherCsmEncryptJobRef	EcucReferenceDef	
BSW Description	Reference to a CSM job to perform the data encryption operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00251]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherCsmEncryptKeyRef		EcucReferenceDef
BSW Description		
Reference to a CSM key associated to the CSM job that performs the data encryption operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00252]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherCsmMacGenerateJobRef		EcucReferenceDef
BSW Description		
Reference to a CSM job to perform the MAC generate operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00258]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherCsmMacGenerateKeyRef		EcucReferenceDef
BSW Description		
Reference to a CSM key associated to the CSM job that performs the MAC generate operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00259]

BSW Module	BSW Context
------------	-------------

Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherCsmMacVerifyJobRef		EcucReferenceDef
BSW Description		
Reference to a CSM job to perform the MAC verify operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00260]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherCsmMacVerifyKeyRef		EcucReferenceDef
BSW Description		
Reference to a CSM key associated to the CSM job that performs the MAC verify operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00261]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	
BSW Parameter		BSW Type
TcplpTlsCipherEncryptKeyLength		EcucIntegerParamDef
BSW Description		
Defines the key length used for en- or decryption. The key length is valid for (symmetric) encryption and decryption.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00253]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker	

BSW Parameter		BSW Type	
TcplpTlsCipherMacKeyLength		EcucIntegerParamDef	
BSW Description			
Specifies the length of the MAC key			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00257]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker		
BSW Parameter		BSW Type	
TcplpTlsCiphersuiteDefinitionRef		EcucReferenceDef	
BSW Description			
Reference to a a ciphersuite definition container			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00250]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsCiphersuite Worker		
BSW Parameter		BSW Type	
TcplpTlsConnectionHandshakeRef		EcucReferenceDef	
BSW Description			
References the container that contains the jobs and keys for handshake operation. Referencing multiple handshake containers allow to share them between workers and to choose the next unused during the handshake.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00249]	

BSW Module	BSW Context		
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites		
BSW Parameter		BSW Type	

TcplpTlsHandshake	EcucParamConfContainerDef
BSW Description	
This container provides information that is needed to process a handshake. It contains the appropriate references to jobs and keys of the CSM to perform the key exchange cryptographic for the ciphersuite and involved certificates.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_Tcplp_00239]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmHashVerifyJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform the hash operation for the whole handshake.	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_Tcplp_00265]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmKeyExchangeCalcPubValJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform the DH Key Exchange algorithm operation	
Template Description	
M2 Parameter	
Mapping Rule	
Mapping Type	
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_Tcplp_00267]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmKeyExchangeCalcSecretJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform the Key Exchange algorithm operation	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00269]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmKeyExchangeDecryptJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform data decryption, e.g. with RSA key exchange operation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00276]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmKeyExchangeDecryptKeyRef	EcucReferenceDef
BSW Description	
Reference to a CSM key to perform data decryption, e.g. with RSA, used for exchange operation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00277]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmKeyExchangeEncryptJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform data encryption, e.g. with RSA key exchange operation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00274]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeEncryptKeyRef	EcucReferenceDef	
BSW Description		
Reference to a CSM key to perform data encryption, e.g. with RSA, used for exchange operation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00275]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeKeyRef	EcucReferenceDef	
BSW Description		
Reference to a CSM key used for Diffie Hellman (DH) key exchange operation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00268]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeSignatureGenerateJobRef	EcucReferenceDef	
BSW Description		
Reference to a CSM job to perform signature generation for DH operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00270]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeSignatureGenerateKeyRef	EcucReferenceDef	
BSW Description	Reference to a CSM key to perform signature generation for DH operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00271]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeSignatureVerifyJobRef	EcucReferenceDef	
BSW Description	Reference to a CSM job to perform signature verification for DH operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00272]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmKeyExchangeSignatureVerifyKeyRef	EcucReferenceDef	
BSW Description	Reference to a CSM key to perform signature verification for DH operation	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00273]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake	
BSW Parameter	BSW Type	
TcplpTlsCsmMasterSecretKeyRef	EcucReferenceDef	
BSW Description		

This is the reference to the master key that is calculated during the session.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00266]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmPRFSupportType	EcucEnumerationParamDef
BSW Description	
Specifies how the CSM job supports the PRF operation.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00264]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmPrfMacJobRef	EcucReferenceDef
BSW Description	
Reference to a CSM job to perform the PRF hash operation	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00262]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsHandshake
BSW Parameter	BSW Type
TcplpTlsCsmPrfMacKeyRef	EcucReferenceDef
BSW Description	
Reference to a CSM key associated to the CSM job that performs the PRF hash operation	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00263]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites	
BSW Parameter	BSW Type	
TcplpTlsPskIdentity	EcucParamConfContainerDef	
BSW Description		
<p>This container provides information about static definition of pre-shared keys. It is used during the handshake to negotiate pre-shared keys between a client and a server.</p> <p>Note: The callbacks for pre-shared keys are an alternative to the static definition. The callbacks allow to define the associated keys at runtime if pre-shared keys are used but no static definition is available. The container definition is used for static configuration.</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00241]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity	
BSW Parameter	BSW Type	
TcplpTlsPresharedKeyCsmKeyRef	EcucReferenceDef	
BSW Description		
Reference to a CSM key associated to the CSM job that performs the PRF hash operation		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00280]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity	
BSW Parameter	BSW Type	
TcplpTlsPresharedKeyIdentity	EcucStringParamDef	
BSW Description		
This item provides the key identification. The TLS client selects the pre-shared key based on the identification hint provided by the server and returns the key identification name back to the server.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00284]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity	
BSW Parameter	BSW Type	
TcplpTlsPresharedKeyIdentityHint	EcucStringParamDef	
BSW Description		
Provides the identity hint for a pre-shared key. This information is transmitted by the TLS Server to provide its identification to the TLS client. The TLS client uses the same information to select the pre-shared key.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00279]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity	
BSW Parameter	BSW Type	
TcplpTlsPskGetClientKeyIdentityFunc	EcucFunctionNameDef	
BSW Description		
Defines the function name for the Up_TlsClientGetPskIdentity() callback.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00281]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity	
BSW Parameter	BSW Type	
TcplpTlsPskGetKeyIdentityHintFunc	EcucFunctionNameDef	
BSW Description		
Defines the function name for the Up_TlsServerGetPskIdentityHint() callback.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Tcplp_00283]
-------	--------------------

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsCiphersuites/TcplpTlsPskIdentity
BSW Parameter	BSW Type
TcplpTlsPskGetServerKeyIdentityFunc	EcucFunctionNameDef
BSW Description	
Defines the function name for the Up_TlsServerGetPskIdentity () callback.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00282]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig
BSW Parameter	BSW Type
TcplpTlsConnection	EcucParamConfContainerDef
BSW Description	
This container defines the properties of a TLS connection	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00223]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection
BSW Parameter	BSW Type
TcplpTlsCertificateIdentityRef	EcucReferenceDef
BSW Description	
References the container that contains the certificate and identity information.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00235]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection

BSW Parameter		BSW Type
TcplpTlsConnectionCiphersuiteWorkerRef		EcucReferenceDef
BSW Description		
References the container that contains the jobs and keys to process the application data.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00234]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsConnectionGetTimeFunc		EcucFunctionNameDef
BSW Description		
Defines the function name for the Up_TlsGetCurrentTimeStamp() callback.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00232]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsConnectionGroupRef		EcucReferenceDef
BSW Description		
Assigns the TLS connection to a connection group.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00233]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsConnectionId		EcucIntegerParamDef
BSW Description		
Identifier of the connection. The set of configured identifiers shall be consecutive and gapless.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00225]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection
BSW Parameter	BSW Type
TcplpTlsConnectionPskIdentityRef	EcucReferenceDef
BSW Description	
References the container that contains information about pre-shared keys.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00236]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection
BSW Parameter	BSW Type
TcplpTlsConnectionType	EcucEnumerationParamDef
BSW Description	
Specifies if the TLS connection is a server or a client.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00226]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection
BSW Parameter	BSW Type
TcplpTlsIpAddressAssignment	EcucReferenceDef
BSW Description	
Contains additional information about the endpoint IP address information. If this reference is present, the IP address of the connecting socket shall also be checked if a TLS connection shall be assigned automatically to a socket.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00229]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsMaxFragmentLength		EcucIntegerParamDef
BSW Description		
Specifies the max length in bytes of a TLS fragment that is sent as a block.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00227]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsPortAssignment		EcucIntegerParamDef
BSW Description		
Specifies the port address that is used for TLS communication.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00285]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter		BSW Type
TcplpTlsUseClientAuthenticationRequest		EcucBooleanParamDef
BSW Description		
Defines if client authentication shall be applied for this TLS connection.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00230]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig/TcplpTlsConnection	
BSW Parameter	BSW Type	
TcplpTlsUseSecurityExtensionRecordSizeLimit	EcucBooleanParamDef	
BSW Description	Defines if the security extension for max_fragment_length shall be supported as defined in IETF RFC 8449, chapter 4.1.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00231]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig	
BSW Parameter	BSW Type	
TcplpTlsConnectionGroup	EcucParamConfContainerDef	
BSW Description	This optional container is used to collect all TlsConnections that belong to a TlsConnectionGroup. The intention of a TLS connection group is to share resources among TLS connections collected in a group, because only one connection of a group can be used at a time.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00224]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig	
BSW Parameter	BSW Type	
TcplpTlsCsmRandomGenerateJobRef	EcucReferenceDef	
BSW Description	Reference to a CSM job to generate a random value.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00221]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpTlsConfig	

BSW Parameter		BSW Type
TcplpTlsMaxConnections		EcucIntegerParamDef
BSW Description		
Defines the max. number of TLS connections that can be opened at the same time.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00220]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig	
BSW Parameter		BSW Type
TcplpUdpConfig		EcucParamConfContainerDef
BSW Description		
Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module		
Template Description		
Content Model for UDP configuration.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::UdpTp		
Mapping Rule		Mapping Type
This container shall be created if the UdpTp element is used in the ECU Extract.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00026]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpConfig/TcplpUdpConfig	
BSW Parameter		BSW Type
TcplpUdpTtl		EcucIntegerParamDef
BSW Description		
Default Time-to-live value of outgoing UDP packets.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00075]

BSW Module	BSW Context	
Tcplp	Tcplp	
BSW Parameter		BSW Type
TcplpGeneral		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack.		

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00002]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral
BSW Parameter	BSW Type
TcplpBufferMemory	EcucIntegerParamDef
BSW Description	
Memory size in bytes reserved for TCP/IP buffers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00016]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral
BSW Parameter	BSW Type
TcplpDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00004]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral
BSW Parameter	BSW Type
TcplpDhcpServerEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Server.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00183]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpGetAndResetMeasurementDataApi		EcucBooleanParamDef
BSW Description		
Enables / Disables the Get and Reset Measurement Data API		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00217]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpV4General		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00163]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter		BSW Type
TcplpArpEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) support of ARP (Address Resolution Protocol).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	

valid	[ECUC_Tcplp_00006]
-------	--------------------

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter	BSW Type	
TcplpAutoIpEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disables (FALSE) the Auto-IP (automatic private IP addressing) sub-module.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00011]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter	BSW Type	
TcplpDhcpClientEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Client.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00010]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter	BSW Type	
TcplpIcmpEnabled	EcucBooleanParamDef	
BSW Description	Enables (TRUE) or disabled (FALSE) support of ICMP (Internet Control Message Protocol).	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00007]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	

BSW Parameter		BSW Type
TcplpV4Enabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) support of IPv4 (Internet Protocol version 4).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00088]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter		BSW Type
TcplpLocalAddrIpv4EntriesMax		EcucIntegerParamDef
BSW Description		
Maximum number of LocalAddr table entries for IPv4.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00018]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral/TcplpV4General	
BSW Parameter		BSW Type
TcplpPathMtuDiscoveryEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) the discovery of the maximum transmission unit on a path according to IETF RfC 1191.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_Tcplp_00012]

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpV6General		EcucParamConfContainerDef
BSW Description		

This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00164]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpDhcpV6ClientEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) Client.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00093]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpV6Enabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of IPv6 (Internet Protocol version 6).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00089]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpV6PathMtuDiscoveryEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) Path MTU Discovery support for IPv6 according to IETF RFC 1981.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00090]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpLocalAddrIpv6EntriesMax	EcucIntegerParamDef
BSW Description	
Maximum number of LocalAddr table entries for IPv6.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00017]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpNdpAddressResolutionUnreachabilityDetection Enabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of Address Resolution and Neighbor Unreachability Detection via NDP.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00091]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral/TcplpV6General
BSW Parameter	BSW Type
TcplpNdpPrefixAndRouterDiscoveryEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of Prefix and Router Discovery via NDP.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00092]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Period of Tcplp_MainFunction in [s].		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00013]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpResetIpAssignmentApi		EcucBooleanParamDef
BSW Description		
Enables/disables the API Tcplp_ResetIpAssignment of a DHCP-client.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00182]	

BSW Module	BSW Context	
Tcplp	Tcplp/TcplpGeneral	
BSW Parameter		BSW Type
TcplpScalabilityClass		EcucEnumerationParamDef
BSW Description		
In order to customize the Tcplp Stack to the specific needs of the user it can be scaled according to the scalability classes.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Tcplp_00169]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpGeneral	
BSW Parameter		BSW Type	
TcplpTcpEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disabled (FALSE) support of TCP (Transmission Control Protocol).			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00008]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpGeneral	
BSW Parameter		BSW Type	
TcplpTcpSocketMax		EcucIntegerParamDef	
BSW Description			
Maximum number of TCP sockets			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00014]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpGeneral	
BSW Parameter		BSW Type	
TcplpUdpEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disabled (FALSE) support of UDP (User Datagram Protocol)			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Tcplp_00009]	

BSW Module		BSW Context	
Tcplp		Tcplp/TcplpGeneral	
BSW Parameter		BSW Type	
TcplpUdpSocketMax		EcucIntegerParamDef	
BSW Description			

Maximum number of UDP sockets.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00015]

BSW Module	BSW Context
Tcplp	Tcplp/TcplpGeneral
BSW Parameter	BSW Type
TcplpVersionInfoApi	EcucBooleanParamDef
BSW Description	
If true the Tcplp_GetVersionInfo API is available.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_Tcplp_00005]

C.6.9 DoIP

BSW Module	BSW Context
DoIP	DoIP
BSW Parameter	BSW Type
DoIPConfigSet	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters and sub containers of the AUTOSAR DoIP module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00003]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet
BSW Parameter	BSW Type
DoIPEid	EcucIntegerParamDef
BSW Description	
Configured EID (Entity ID of) for vehicle identification/vehicle announcement. Only necessary if DoIPUseMacAddressForIdentification is set to FALSE.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00014]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet
BSW Parameter	BSW Type
DoIPGid	EcucIntegerParamDef
BSW Description	
Configured GID (Group ID of) for vehicle identification/vehicle announcement.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00015]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet
BSW Parameter	BSW Type
DoIPInterface	EcucParamConfContainerDef
BSW Description	
This container defines a logical IP interface and collects properties to configure this interface.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00100]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface
BSW Parameter	BSW Type
DoIPAliveCheckResponseTimeout	EcucFloatParamDef
BSW Description	
Timeout in [s] for waiting for a response to an Alive Check request before the connection is considered to be disconnected. Represents parameter T_TCP_AliveCheck of ISO 13400-2:2012.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00009]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPChannel		EcucParamConfContainerDef
BSW Description		
Configuration of one DoIPChannel.		
Template Description		
A connection identifies the sender and the receiver of this particular communication. The DoIP module routes a tpSdu through this connection.		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DoIPtpConnection		
Mapping Rule		Mapping Type
One DoIPChannel container is created for each DoIP channel of the configured EcuInstance. A DoIP channel is constituted by the set of all DoIPtpConnection elements via which the configured EcuInstance sends or receives SDUs (SDUs are referenced by the PduTriggering DoIPtpConnection.tpSdu refers to) and that share the same local diagnosis address and tester address.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00069]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel	
BSW Parameter		BSW Type
DoIPChannelSARef		EcucReferenceDef
BSW Description		
Reference to the DoIPTester.		
Template Description		
Reference to the address of the sender of the tpSdu.		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DoIPtpConnection.dolpSourceAddress		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00070]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel	
BSW Parameter		BSW Type
DoIPChannelTARef		EcucReferenceDef
BSW Description		
Reference to the target address.		
Template Description		
Reference to the address of the receiver of the tpSdu.		
M2 Parameter		

SystemTemplate::DiagnosticConnection::DoIPtpConnection.dolpTargetAddress	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00071]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel
BSW Parameter	BSW Type
DoIPPduRRxPdu	EcucParamConfContainerDef
BSW Description	
This container contains the Rx Pdus to connect with the Rx Pdus of the PduR.	
Template Description	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DoIPtpConnection.tpSdu	
Mapping Rule	Mapping Type
If the DoIP channel receives an SDU, one DoIPPduRRxPdu container is created.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00055]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRRxPdu
BSW Parameter	BSW Type
DoIPPduRRxPduld	EcucIntegerParamDef
BSW Description	
The DoIPPduRRxPduld is required by the API call DoIP_TpCancelReceive.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00057]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRRxPdu
BSW Parameter	BSW Type
DoIPPduRRxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Reference to the EcuC container that corresponds to the SDU received by the DoIP channel	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00058]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel	
BSW Parameter	BSW Type	
DoIPPduRTxPdu	EcucParamConfContainerDef	
BSW Description		
This container contains the Tx Pdus to connect with the Tx Pdus of the PduR. If the parameter is not configured the channel is for functional addressing.		
Template Description		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DoIPtpConnection.tpSdu		
Mapping Rule	Mapping Type	
If the DoIP channel sends an SDU, one DoIPPduRTxPdu container is created.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00056]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRTxPdu	
BSW Parameter	BSW Type	
DoIPPduRTxPduld	EcucIntegerParamDef	
BSW Description		
The DoIPPduRTxPduld is required by DoIP_TpTransmit or DoIP_IfTransmit and DoIP_TpCancelTransmit.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00060]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRTxPdu	
BSW Parameter	BSW Type	
DoIPPduRTxPduRef	EcucReferenceDef	
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Template Description		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DoIPtpConnection.tpSdu		
Mapping Rule	Mapping Type	
Reference to the EcuC container that corresponds to the SDU sent by the DoIP channel	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00059]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRTxPdu	
BSW Parameter		BSW Type	
DoIPPduType		EcucEnumerationParamDef	
BSW Description			
API Type to use for communication with PduR. DOIP_IFPDU for UUDT messages, DOIP_TPPDU for all other diagnostic messages.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00075]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type	
DoIPConnections		EcucParamConfContainerDef	
BSW Description			
Container contains all lower layer connection specific information, i.e. the single Pdu References and Handle IDs to the SoAd.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00032]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections	
BSW Parameter		BSW Type	
DoIPTargetAddress		EcucParamConfContainerDef	
BSW Description			
This container describes a possible TargetAddress that is supported by DoIP.			
Template Description			
The logical DoIP address.			
M2 Parameter			
SystemTemplate::TransportProtocols::DoIpLogicAddress			
Mapping Rule			Mapping Type
This container shall be created for each DoIpLogicAddress referenced by a DoIpTpConnection in the role doIpTargetAddress			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00053]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTargetAddress	

BSW Parameter		BSW Type
DoIPTargetAddressValue		EcucIntegerParamDef
BSW Description		
Valid Target Address of a DoIP target address.		
Template Description		
The logical DoIP address.		
M2 Parameter		
SystemTemplate::TransportProtocols::DoIPLogicAddress.address		
Mapping Rule		Mapping Type
This value shall be derived from DoIPLogicAddress that is referenced by DoIP Connection in the role doIPTargetAddress		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00054]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections	
BSW Parameter		BSW Type
DoIPTcpConnection		EcucParamConfContainerDef
BSW Description		
This container describes a TCP connection to the lower layer SoAd module.		
Template Description		
Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection		
Mapping Rule		Mapping Type
A DoIPTcpConnection is created for each SocketConnection for which the following conditions hold: - Protocol: Tcp, i.e. the SocketConnection refers to a local SocketAddress containing an ApplicationEndpoint which in turn contains a TcpTp configuration.- DoIP PDU transmission: The EcuInstance sends and/or receives via the Socket Connection PDUs that are of Type GeneralPurposeIPdu and have their category set to "DoIP".		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00045]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection	
BSW Parameter		BSW Type
DoIPRequestAddressAssignment		EcucBooleanParamDef
BSW Description		
The DoIP module shall request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcpIpLocalAddr related to this DoIPConnection.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00095]

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection	
BSW Parameter		BSW Type	
DoIPSoAdTcpRxPdu		EcucParamConfContainerDef	
BSW Description			
This container describes a Rx PDU received via SoAd over TCP			
Template Description			
Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier			
Mapping Rule			Mapping Type
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is received via the StaticSocketConnection that represents the DoIPTcpConnection.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00080]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpRxPdu	
BSW Parameter		BSW Type	
DoIPSoAdTcpRxPdulId		EcucIntegerParamDef	
BSW Description			
The DoIPSoAdTcpRxPdulId is required by the API call DoIP_SoAdTpRxIndication to receive I-PDUs from the SoAd.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00082]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpRxPdu	
BSW Parameter		BSW Type	
DoIPSoAdTcpRxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
Template Description			
Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier			
Mapping Rule			Mapping Type
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu receiving by the DoIPTcpConnection.			full
Mapping Status		ECUC Parameter ID	
obsolete		[ECUC_DoIP_00083]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection	
BSW Parameter		BSW Type	
DoIPSoAdTcpTxPdu		EcucParamConfContainerDef	
BSW Description			
This container describes a Tx PDU sent via SoAd over TCP			
Template Description			
Reference to a Pdu that is transmitted over a socket connection.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::SoConIPdulIdentifier.pduTriggering			
Mapping Rule			Mapping Type
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is transmitted via the StaticSocketConnection that represents the DoIPTcpConnection.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00081]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpTxPdu	
BSW Parameter		BSW Type	
DoIPSoAdTcpTxPdulId		EcucIntegerParamDef	
BSW Description			
The DoIPSoAdTcpTxPdulId is required by the API call DoIP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00085]	

BSW Module		BSW Context	
DoIP		DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpTxPdu	
BSW Parameter		BSW Type	
DoIPSoAdTcpTxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
Template Description			
Assignment of IPdulIdentifiers that are transmitted over the static SocketConnection.			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier			
Mapping Rule			Mapping Type
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the DoIPTcpConnection.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_DoIP_00084]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection	
BSW Parameter	BSW Type	
DoIPTcpConnectionSecurityRequired	EcucBooleanParamDef	
BSW Description	Indicates if the associated TCP socket uses a secure connection (e.g. TLS)	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00097]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections	
BSW Parameter	BSW Type	
DoIPUdpConnection	EcucParamConfContainerDef	
BSW Description	This Container describes a Udp connection to the lower layer SoAd module.	
Template Description	Assignment of IPduldentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPduldentifier	
Mapping Rule	Mapping Type	
A DoIPUdpConnection is created for each StaticSocketConnection for which the following conditions hold: <ul style="list-style-type: none"> * Protocol: Udp, i.e. the StaticSocketConnection is aggregated by the local SocketAddress containing an ApplicationEndpoint which in turn contains a Udp Tp configuration. * DoIP PDU transmission: The EcuInstance sends and/or receives via the SocketConnection PDUs that are of Type GeneralPurposeIPdu and have their category set to "DoIP". 		
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00052]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection	
BSW Parameter	BSW Type	
DoIPRequestAddressAssignment	EcucBooleanParamDef	
BSW Description	The DoIP module shall request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcplpLocalAddr related to this DoIPConnection.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00095]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection	
BSW Parameter	BSW Type	
DoIPSoAdUdpRxPdu	EcucParamConfContainerDef	
BSW Description	This container describes a Rx PDU received via SoAd over UDP.	
Template Description	Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier	
Mapping Rule	Mapping Type	
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is received via the StaticSocketConnection that represents the DoIPUdpConnection.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00046]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpRxPdu	
BSW Parameter	BSW Type	
DoIPSoAdUdpRxPdulId	EcucIntegerParamDef	
BSW Description	The DoIPSoAdUdpRxPdulId is required by the API call DoIP_SoAdIfRxIndication to receive I-PDUs from the SoAd.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00048]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpRxPdu	
BSW Parameter	BSW Type	
DoIPSoAdUdpRxPduRef	EcucReferenceDef	
BSW Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier	
Mapping Rule	Mapping Type	
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu received by the DoIPUdpConnection.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00049]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection	
BSW Parameter		BSW Type
DoIPSoAdUdpTxPdu		EcucParamConfContainerDef
BSW Description		
This container describes a Tx PDU sent via SoAd over UDP.		
Template Description		
Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier		
Mapping Rule		Mapping Type
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is transmitted via the StaticSocketConnection that represents the DoIPUdpConnection.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00047]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpTxPdu	
BSW Parameter		BSW Type
DoIPSoAdUdpTxPduld		EcucIntegerParamDef
BSW Description		
The DoIPSoAdUdpTxPduld is required by the API call DoIP_SoAdIfTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00051]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpTxPdu	
BSW Parameter		BSW Type
DoIPSoAdUdpTxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Template Description		
Assignment of IPdulidentifiers that are transmitted over the static SocketConnection.		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPdulIdentifier		
Mapping Rule		Mapping Type
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the DoIPUdpConnection.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00050]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections	
BSW Parameter	BSW Type	
DoIPUdpVehicleAnnouncementConnection	EcucParamConfContainerDef	
BSW Description	This container describes the UDP multicast connections to the lower layer SoAd module.	
Template Description	Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.	
M2 Parameter	SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection	
Mapping Rule	Mapping Type	
<p>This container shall be created:</p> <ul style="list-style-type: none"> - if a StaticSocketConnection contains a single SoConIpdIdentifier that references a PduTriggering of a GeneralPurposePdu with category set to DoIp - if the GeneralPurposePdu with category set to DoIp is sent by the regarded ECU. - if the SocketAddress containing this StaticSocketConnection contains an ApplicationEndpoint with a UdpTp configuration - if the SocketAddress containing this SocketConnection references contains an ipAddress that either is a the limited broadcast address (i.e., 255.255.255.255) in case of IPv4 or the link-local scope multicast address (i.e., FF02::1) in case of IPv6. 	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00076]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection	
BSW Parameter	BSW Type	
DoIPRequestAddressAssignment	EcucBooleanParamDef	
BSW Description	The DoIP module shall request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcpIpLocalAddr related to this DoIpConnection.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00095]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection	
BSW Parameter	BSW Type	
DoIPSoAdUdpVehicleAnnouncementTxPdu	EcucParamConfContainerDef	
BSW Description	This container describes the vehicle announcement TxPdu sent via the SoAd.	
Template Description		

Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remoteAddress.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection	
Mapping Rule	Mapping Type
This container shall be created: - if a StaticSocketConnection contains a single SoConIpduIdentifier that references a PduTriggering of a GeneralPurposePdu with category set to DoIp - if the GeneralPurposePdu with category set to DoIp is sent by the regarded ECU. - if the SocketAddress containing this SocketConnection contains an Application Endpoint with a UdpTp configuration - if the SocketAddress containing this SocketConnection contains an ipAddress that either is a the limited broadcast address (i.e., 255.255.255.255) in case of IPv4 or the link-local scope multicast address (i.e., FF02::1) in case of IPv6.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00077]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection/DoIPSoAdUdpVehicleAnnouncementTxPdu
BSW Parameter	BSW Type
DoIPSoAdUdpVehicleAnnouncementTxPduId	EcucIntegerParamDef
BSW Description	
The DoIPSoAdUdpVehicleAnnouncementTxPduId is required by the API call DoIP_SoAdIfTxConfirmation() that is called by the SoAd to confirm that the IPdu has been transmitted successfully.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00078]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection/DoIPSoAdUdpVehicleAnnouncementTxPdu
BSW Parameter	BSW Type
DoIPSoAdUdpVehicleAnnouncementTxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances::StaticSocketConnection.iPduIdentifier	
Mapping Rule	Mapping Type
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the DoIPUdpVehicleAnnouncement connection.	local
Mapping Status	ECUC Parameter ID

valid	[ECUC_DoIP_00079]
-------	-------------------

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPFurtherActionByteCallback	EcucParamConfContainerDef	
BSW Description		
This container describes the Callbackfunction to get the Further Action byte. This container shall always be present. If the DoIPFurtherActionByteDirect parameter is not present, the DoIP module will use an RPort of ServiceInterface CallbackGetFurtherActionByte with the name "CBGetFurtherActionByte_<shortname of enclosing DoIPInterface container>".		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00092]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPFurtherActionByteCallback	
BSW Parameter	BSW Type	
DoIPFurtherActionByteDirect	EcucFunctionNameDef	
BSW Description		
Direct C Callback function to get the OEM specific Further Action Byte for the DoIP vehicle identification response/vehicle announcement. If the DoIPFurtherActionByteDirect parameter is present, the DoIP module will not use an RPort of ServiceInterface "CBGetFurtherActionByte" but will call the configured function.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00093]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPGeneralInactivityTime	EcucFloatParamDef	
BSW Description		
Timeout in [s] for maximum inactivity of a TCP socket connection before the DoIP module will close the according socket connection. Represents parameter T_TCP_General_Inactivity of ISO 13400-2:2012		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00068]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPInitialInactivityTime		EcucFloatParamDef
BSW Description		
Timeout in [s] used for initial inactivity of a connected TCP socket connection directly after socket connection. Represents parameter T_TCP_Initial_Inactivity of ISO 13400-2:2012		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00010]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPInitialVehicleAnnouncementTime		EcucFloatParamDef
BSW Description		
Time to wait in [s] for sending first vehicle announcement message after IP address assignment. Represents parameter A_DoIP_Announce_Wait of ISO 13400-2:2012		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00008]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPInterfaceActLineCtrl		EcucBooleanParamDef
BSW Description		
This attribute defines whether the network interface - is started "on-demand" when an activation line is sensed (TRUE) or - is always available (FALSE).		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00101]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPInterfaceAnnouncementStart	EcucEnumerationParamDef	
BSW Description		
This attribute defines, when vehicle announcement is started on a DoIPInterface		
* Automatic: As soon as the underlying UDP vehicle announcement connection switches to SOAD_SOCON_ONLINE		
* OnTrigger: As soon as the API DoIP_TriggerVehicleAnnouncement is called for the given DoIPInterface instance		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00099]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPInterfaceId	EcucIntegerParamDef	
BSW Description		
This parameter is an identifier of the DoIPInterface. The value of this parameter will be assigned to the symbolic name derived from the container short name.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00098]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPMaxTesterConnections	EcucIntegerParamDef	
BSW Description		
Maximum amount of tester connections that shall be maintained at one time before alive check is performed.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00012]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPRoutingActivation		EcucParamConfContainerDef
BSW Description		
This container describes the routing activation possibilities by representing for each container a possible routing activation request message to the DoIP entity and the according references to the activated diagnostic messages.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00030]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation	
BSW Parameter		BSW Type
DoIPRoutingActivationAuthenticationCallback		EcucParamConfContainerDef
BSW Description		
Container describes the Callbackfunction to call on a Routing Activation Request for Authentication. If this container is configured but the DoIPRoutingActivationAuthenticationFunc parameter is not present, the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActivation container.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00035]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback	
BSW Parameter		BSW Type
DoIPRoutingActivationAuthenticationFunc		EcucFunctionNameDef
BSW Description		
Direct C Callback function to trigger the authentication function for routing activation. If the DoIPRoutingActivationAuthenticationFunc parameter is present, the DoIP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00039]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback
BSW Parameter	BSW Type
DoIPRoutingActivationAuthenticationReqLength	EcucIntegerParamDef
BSW Description	
Describes the amount of bytes used to handle to the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationReqData will not be handled to the API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00040]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback
BSW Parameter	BSW Type
DoIPRoutingActivationAuthenticationResLength	EcucIntegerParamDef
BSW Description	
Describes the amount of bytes used to read by the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationResData will not be fetched via the API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00041]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation
BSW Parameter	BSW Type
DoIPRoutingActivationConfirmationCallback	EcucParamConfContainerDef
BSW Description	

Container describes the Callbackfunction to call on a Routing Activation Request for Confirmation. If this container is configured but the DoIPRoutingActivationConfirmationFunc parameter is not present the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActivation container.

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00061]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback
BSW Parameter	BSW Type
DoIPRoutingActivationConfirmationFunc	EcucFunctionNameDef
BSW Description	
Direct C Callback function to trigger the confirmation function for routing activation. If the DoIPRoutingActivationConfirmationFunc parameter is present the DoIP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00036]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback
BSW Parameter	BSW Type
DoIPRoutingActivationConfirmationReqLength	EcucIntegerParamDef
BSW Description	
Describes the amount of bytes used to handle to the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedReqData will not be handled to the API.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00037]

BSW Module	BSW Context
-------------------	--------------------

DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback	
BSW Parameter		BSW Type
DoIPRoutingActivationConfirmationResLength		EcucIntegerParamDef
BSW Description		
Describes the amount of bytes used to read by the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedResData will not be fetched via the API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00038]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation	
BSW Parameter		BSW Type
DoIPRoutingActivationNumber		EcucIntegerParamDef
BSW Description		
Identifies the Routing activation Number which is received for a DoIP routing activation request message.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00033]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation	
BSW Parameter		BSW Type
DoIPRoutingActivationSecurityRequired		EcucBooleanParamDef
BSW Description		
Indicates if a routing activation requires a secure TCP connection		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00096]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation	
BSW Parameter		BSW Type

DoIPTargetAddressRef	EcucReferenceDef
BSW Description	
Reference to all DoIPTargetAddress which are activated on this Routing activation.	
Template Description	
M2 Parameter	
Mapping Rule	
	Mapping Type
	local
Mapping Status	
valid	ECUC Parameter ID
	[ECUC_DoIP_00034]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPTester		EcucParamConfContainerDef
BSW Description		
This container describes the properties of the possible connectable Tester for the DoIP entity.		
Template Description		
A connection identifies the sender and the receiver of this particular communication. The Dolp module routes a tpSdu through this connection.		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DolpTpConnection		
Mapping Rule		Mapping Type
One DoIPTester container is created for each valid and unique tester address value of any DolpTpConnection of the configured EcucInstance. If the configured EcucInstance receives the PDU Triggering referenced via DolpTpConnection.tpSdu, the tester address of a DolpTpConnection is referenced via DolpLogicAddress.dolpSourceAddress. If the configured EcucInstance sends the PDU Triggering referenced via DolpTpConnection.tpSdu, the tester address of a DolpTpConnection is referenced via DolpLogicAddress.dolpTargetAddress.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00031]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPTester	
BSW Parameter		BSW Type
DoIPNumByteDiagAckNack		EcucIntegerParamDef
BSW Description		
Specifies the number of original Diagnostic request bytes the DoIP entity responses on a NACK of a diagnostic response message to the Tester.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00042]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPTester	
BSW Parameter		BSW Type
DoIPRoutingActivationRef		EcucReferenceDef
BSW Description		
Reference to a DoIPRoutingActivation describing the possible routing activations of the DoIPTester		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00062]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPTester	
BSW Parameter		BSW Type
DoIPTesterSA		EcucIntegerParamDef
BSW Description		
Source Address of the Tester sent via routing activation or diagnostic message.		
Template Description		
The logical DoIP address.		
M2 Parameter		
SystemTemplate::TransportProtocols::DolpLogicAddress.address		
Mapping Rule		Mapping Type
<p>If the configured EcucInstance receives the PDU Triggering referenced via DolpTpConnection.tpSdu, the tester address of a DolpTpConnection is referenced via DolpLogicAddress.dolpSourceAddress.</p> <p>If the configured EcucInstance sends the PDU Triggering referenced via DolpTpConnection.tpSdu, the tester address of a DolpTpConnection is referenced via DolpLogicAddress.dolpTargetAddress.</p>		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00043]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		BSW Type
DoIPUseMacAddressForIdentification		EcucBooleanParamDef
BSW Description		
<p>Provided the information if a configured EID at vehicle identification response/vehicle announcement is used or the MAC address.</p> <p>TRUE: Use MAC Address instead of EID for Vehicle identification/announcement.</p> <p>FALSE: Use configured EID for vehicle identification/announcement. Dependencies: DoIPEID</p>		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00013]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPUseVehicleIdentificationSyncStatus	EcucBooleanParamDef	
BSW Description		
Defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00016]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPVehicleAnnouncementCount	EcucIntegerParamDef	
BSW Description		
Number of vehicle announcement messages on IP address assignment. Represents parameter A_DoIP_Announce_Num of ISO 13400-2:2012.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00094]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter	BSW Type	
DoIPVehicleAnnouncementInterval	EcucFloatParamDef	
BSW Description		
Time to wait in [s] for sending subsequent vehicle announcement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_DoIP_00007]	

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet	

BSW Parameter		BSW Type
DoIPLogicalAddress		EcucIntegerParamDef
BSW Description		
Describes the logical address of the DoIP entity, i.e. the LA that will route diagnostic requests to the Dcm of the DoIP entity.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00020]

BSW Module	BSW Context	
DoIP	DoIP	
BSW Parameter		BSW Type
DoIPGeneral		EcucParamConfContainerDef
BSW Description		
This container specifies the general configuration parameters of the DoIP module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00002]

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		BSW Type
DoIPDevelopmentErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_DoIP_00004]

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		BSW Type
DoIPDhcpOptionVinUse		EcucBooleanParamDef

BSW Description	
If DoIPDhcpOptionVinUse is set to true the DoIP module will add the VIN to the Dhcp host name if no valid Dhcp host name is already set.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00067]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPEntityStatusMaxByteFieldUse	EcucBooleanParamDef
BSW Description	
This parameter is used to distinguish the optional support of the Max data size element of a diagnostic entity status response.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00064]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPGIDInvalidityPattern	EcucIntegerParamDef
BSW Description	
Specifies the Byte pattern that is used for response messages if no valid GID could be retrieved. Only the value '0' or '255' is allowed".	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00065]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPGetGidCallback	EcucParamConfContainerDef
BSW Description	

This container describes the usage of a callback function to get the GID. (If this container is not present no callback function shall be used by DoIP module to retrieve the GID.)	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00024]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral/DoIPGetGidCallback
BSW Parameter	BSW Type
DoIPGetGidDirect	EcucFunctionNameDef
BSW Description	
If the DoIPGetGidDirect parameter exist the DoIP module shall call the configured callback function (<User>_DoIPGetGID) direct. (It is not needed to specify a service port to the DoIP service component.)	
If the DoIPGetGidDirect parameter does NOT exist the DoIP module shall use a RPort with a Call-backGetGID type of client-server port interface to retrieve the GID.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00028]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPHostNameSizeMax	EcucIntegerParamDef
BSW Description	
Maximum Size of the DHCP HostName in ASCII. This parameter is necessary to reserve the correct amount of bytes for working with the DHCP HostName option. Minimum range is 5 because Dhcp Host Name should be at least "DoIP-" on any configuration.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00073]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type

DoIPMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Determines the frequency at which the DoIP_MainFunction() is called in [s].	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00006]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPMaxRequestBytes	EcucIntegerParamDef
BSW Description	
Specifies the maximum allowed bytes of a DoIP message request without the DoIP header.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00019]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPMaxUDPRequestPerMessage	EcucIntegerParamDef
BSW Description	
This parameter captures the maximum amount of UDP Requests necessary to handle parallel within a single UDP connection.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00074]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPNodeType	EcucEnumerationParamDef
BSW Description	
Describes the Type of the DoIP node.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00021]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPPowerModeCallback	EcucParamConfContainerDef
BSW Description	
This container describes the usage of a callback function to retrieve the current power mode. This container shall always be present.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00023]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral/DoIPPowerModeCallback
BSW Parameter	BSW Type
DoIPPowerModeDirect	EcucFunctionNameDef
BSW Description	
If the DoIPPowerModeDirect parameter exist the DoIP module shall call the configured callback function (<User>_DoIPGetPowerModeCallback) direct. (It is not needed to specify a service port to the DoIP service component.)	
If the DoIPPowerModeDirect parameter does NOT present the DoIP module shall use a RPort with a CallbackGetPowerMode type of client-server port interface to retrieve the current power mode.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00027]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPTriggerGidSyncCallback	EcucParamConfContainerDef
BSW Description	
This container describes the usage of a callback function to trigger the GID synchronization. (If this container does not exist no callback function shall be used by DoIP module to trigger the GID synchronization.)	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00025]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral/DoIPTriggerGidSyncCallback
BSW Parameter	BSW Type
DoIPTriggerGidSyncDirect	EcucFunctionNameDef
BSW Description	
<p>If the DoIPTriggerGidSyncDirect parameter exist the DoIP module shall call the configured callback function (<User>_DoIPTriggerGidSyncCallback) direct. (It is not needed to specify a service port to the DoIP service component.)</p> <p>If the DoIPTriggerGidSyncDirect parameter does NOT present the DoIP module shall use a RPort with a CallbackTriggerGIDSynchnonization type of client-server port interface to trigger the GID synchronization.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00029]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPUseEIDasGid	EcucBooleanParamDef
BSW Description	
<p>Specifies if the DoIP entity shall use its EID if it is the Master for vehicle identification gid on the vehicle identification/vehicle announcement.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00018]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPVersionInfoApi	EcucBooleanParamDef
BSW Description	

<p>Activates the DoIP_GetVersionInfo() API. TRUE: Enables the DoIP_GetVersionInfo() API. FALSE: DoIP_GetVersionInfo() API is not included.</p>	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00005]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPVinGidMaster	EcucBooleanParamDef
BSW Description	
Specifies if the DoIP entity is the Vehicle identification Master for the GID (Group ID).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00017]

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	BSW Type
DoIPVinInvalidityPattern	EcucIntegerParamDef
BSW Description	
Specifies the Byte pattern that is used for response messages if no valid VIN could be retrieved.	
Only the value '0' or '255' is allowed".	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_DoIP_00066]

C.6.10 UdpNm

BSW Module	BSW Context
UdpNm	UdpNm
BSW Parameter	BSW Type
UdpNmGlobalConfig	EcucParamConfContainerDef

BSW Description	
This container contains all global configuration parameters of UDP NM configured from the CanTrcv Module perspective.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00001]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	BSW Type
UdpNmBusSynchronizationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling bus synchronization support.	
This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmBusSynchronizationEnabled.	
Template Description	
Enables bus synchronization support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00006]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	BSW Type
UdpNmChannelConfig	EcucParamConfContainerDef
BSW Description	
This container contains the channel-specific configuration parameters of the UdpNm.	
Template Description	
Udp specific NmCluster attributes	
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00017]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmActiveWakeupBitEnabled	EcucBooleanParamDef

BSW Description	
Enables/Disables the handling of the Active Wakeup Bit in the UdpNm module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00074]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmAllNmMessagesKeepAwake	EcucBooleanParamDef
BSW Description	
Specifies if UdpNm drops irrelevant NM PDUs.	
false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling	
true: Every NM PDU triggers the standard RX indication handling	
Template Description	
Specifies if Nm drops irrelevant NM PDUs.	
false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake	
true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake	
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmNode.allNmMessagesKeepAwake	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00089]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmCarWakeUpBitPosition	EcucIntegerParamDef
BSW Description	
Specifies the Bit position of the CWU within the NM PDU.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00087]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmCarWakeUpBytePosition	EcucIntegerParamDef	
BSW Description	Specifies the Byte position of the CWU within the NM PDU.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00086]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmCarWakeUpFilterEnabled	EcucBooleanParamDef	
BSW Description	If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier UdpNmCarWakeUpFilterNodeId is considered as CWU request. FALSE - CWU filtering is not supported TRUE - CWU filtering is supported.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00077]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmCarWakeUpFilterNodeId	EcucIntegerParamDef	
BSW Description	Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier UdpNmCarWakeUpFilterNodeId is considered as CWU request.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00078]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmCarWakeUpRxEnabled	EcucBooleanParamDef	
BSW Description		
Enables or disables support of CarWakeUp bit evaluation in received NM PDUs.		
FALSE - CarWakeUp not supported. TRUE - CarWakeUp supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00076]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmComMNetworkHandleRef	EcucReferenceDef	
BSW Description		
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00018]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmImmediateNmCycleTime	EcucFloatParamDef	
BSW Description		
Defines the immediate NM PDU cycle time in seconds which is used for UdpNmImmediateNmTransmissions NM PDU transmissions.		
Template Description		
Defines the immediate NmPdu cycle time in seconds which is used for nmlImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmlImmediateNmTransmissions is greater one.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmlImmediateNmCycleTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00079]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmImmediateNmTransmissions	EcucIntegerParamDef	
BSW Description		
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immediate NM PDUs is defined by UdpNmImmediateNmCycleTime.		
Template Description		
Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmlmmediateNmCycleTime.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmlmmediateNmTransmissions		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00075]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
Call cycle of UdpNm_MainFunction_x for the respective instance in [s].		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00032]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmMsgCycleOffset	EcucFloatParamDef	
BSW Description		
Time offset in the periodic transmission node. It determines the start delay of the transmission.		
< UdpNmMsgCycleTime		
This parameter is only valid if UdpNmPassiveModeEnabled is disabled.		
Template Description		
Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmNode.nmMsgCycleOffset		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00029]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmMsgCycleTime	EcucFloatParamDef	
BSW Description		
Period of a NM-message. It determines the periodic rate and is the basis for transmit scheduling. $NmTimeoutTime = n * UdpNmMsgCycleTime$ This parameter is only valid if UdpNmPassiveModeEnabled is disabled.		
Template Description		
Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmMsgCycleTime		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00028]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmNodeDetectionEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling the node detection support. This parameter shall be derived from NmNodeDetectionEnabled. This parameter shall only be enabled if UdpNmNodeIdEnabled == true. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False).		
Template Description		
Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmNodeDetectionEnabled		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00090]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmNodeId	EcucIntegerParamDef	
BSW Description		

Node identifier of local node.	
This parameter is only valid if UdpNmPassiveModeEnabled is set to OFF and UdpNmNodeDetectionEnabled is set to ON.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmNodeId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00031]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmNodeIdEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the source node identifier.	
This parameter shall be derived from NmNodeIdEnabled.	
Template Description	
Enables the source node identifier.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmNodeIdEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00091]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmPduCbvPosition	EcucEnumerationParamDef
BSW Description	
Defines the position of the control bit vector within the NM PACKET.	
The value of the parameter represents the location of the control bit vector in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means the control bit vector is not part of the NM PACKET)	
See also UdpNmPduNidPosition	
<pre>if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition != UDPNM_PDU_OFF) then UdpNmPduCbvPosition != UdpNmPduNidPosition</pre>	
<pre>if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition == UDPNM_PDU_OFF) then UdpNmPduCbvPosition = UDPNM_PDU_BYTE0</pre>	
Template Description	
Defines the position of the control bit vector within the NmPdu (Byte position).	
M2 Parameter	

SystemTemplate::NetworkManagement::UdpNmCluster.nmCbvPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00026]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmPduNidPosition	EcucEnumerationParamDef
BSW Description	
<p>Defines the position of the source node identifier within the NM PACKET.</p> <p>ImplementationType: UdpNm_PduPositionType</p> <p>The value of the parameter represents the location of the source node identifier in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means source node identifier is not part of the NM PACKET)</p> <p>See also UdpNmPduCbvPosition if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_NID_POSITION != UDPNM_PDU_CBV_POSITION</p> <p>if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_IND_POSITION = UDPNM_PDU_BYTE0</p>	
Template Description	
Defines the byte position of the source node identifier within the NmPdu.	
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmNidPosition	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00025]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmPnEnabled	EcucBooleanParamDef
BSW Description	
<p>Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported</p>	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmPncParticipation	
Mapping Rule	Mapping Type
If NmCluster.nmPncParticipation has the value "true" or is not defined then Udp NmPnEnabled shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00061]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmPnEraCalcEnabled	EcucBooleanParamDef	
BSW Description		
Specifies if UdpNm calculates the PN request information for external requests. (ERA) false: PN request are not calculated true: PN request are calculated.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00060]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmPnEraRxNSduRef	EcucReferenceDef	
BSW Description		
Reference to a Pdu in the COM-Stack. The SduRef is required for every UdpNm Channel, because ERA is reported per channel.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00073]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmPnHandleMultipleNetworkRequests	EcucBooleanParamDef	
BSW Description		
false: UdpNm_NetworkRequest is ignored in NO. true: UdpNm_NetworkRequest triggers a change from NO to RM.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00063]

BSW Module	BSW Context

UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmRemoteSleepIndTime	EcucFloatParamDef	
BSW Description		
Timeout for Remote Sleep Indication. It defines the time in [s] how long it shall take to recognize that all other nodes are ready to sleep. Typically it should be equal to: $n * \text{UdpNmMsgCycleTime}$, where n denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the Remote Sleep Indication procedure.		
Template Description		
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmRemoteSleepIndicationTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00023]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmRepeatMessageTime	EcucFloatParamDef	
BSW Description		
Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.		
Template Description		
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.		
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmRepeatMessageTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00022]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmRepeatMsgIndEnabled	EcucBooleanParamDef	
BSW Description		
Enable/disable the notification that a RepeatMessageRequest bit has been received.		
Template Description		
Switch for enabling the Repeat Message Bit Indication.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmRepeatMsgIndEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00092]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmRetryFirstMessageRequest		EcucBooleanParamDef
BSW Description		
Specifies if first message request in UdpNm is repeated until accepted by SoAd.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00085]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmRxpDdu		EcucParamConfContainerDef
BSW Description		
This container describes the UdpNm RX PDU's.		
Template Description		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.rxNmPdu		
Mapping Rule		Mapping Type
Create container for each NmPdu that is received on the regarded Nm cluster		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00038]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxpDdu	
BSW Parameter		BSW Type
UdpNmRxpDduId		EcucIntegerParamDef
BSW Description		
ID of the RxPdu that will be used by a RxIndication of the lower layer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00043]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxpDdu	
BSW Parameter		BSW Type
UdpNmRxpDduRef		EcucReferenceDef
BSW Description		

The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00039]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmStayInPbsEnabled	EcucBooleanParamDef
BSW Description	
If this parameter is disabled Prepare Bus-Sleep Mode is left after UdpNmWaitBusSleepTime. If this parameter is enabled Prepare Bus-Sleep Mode can only be left if ECU is powered off or any restart reason applies.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00093]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	BSW Type
UdpNmTimeoutTime	EcucFloatParamDef
BSW Description	
Network Timeout for NM packets. It denotes the time in [s] how long the NM shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be greater than UdpNmMsgCycleTime. Typically, it should be equal to: $x * \text{UdpNmMsgCycleTime}$, where x denotes the number of NM PACKET cycle times in the Ready Sleep State before transition into the Bus-Sleep Mode is initiated. The value of x decremented by one determines the amount of lost NM packets that can be tolerated by the coordination algorithm.	
Template Description	
Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.	
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmNetworkTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_UdpNm_00020]
-------	--------------------

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	BSW Type	
UdpNmTxPdu	EcucParamConfContainerDef	
BSW Description		
This container describes the UdpNm TX PDU's.		
Template Description		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule		Mapping Type
Create container for each NmPdu that is transmitted on the regarded Nmcluster		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00036]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmTxPdu	
BSW Parameter	BSW Type	
UdpNmTxConfirmationPduId	EcucIntegerParamDef	
BSW Description		
Id of the TxPdu that will be used by a TxConfirmation from the lower layer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00042]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmTxPdu	
BSW Parameter	BSW Type	
UdpNmTxPduRef	EcucReferenceDef	
BSW Description		
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00037]

BSW Module	BSW Context

UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type
UdpNmUserDataTxPdu	EcucParamConfContainerDef	
BSW Description		
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.		
Template Description		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		Mapping Type
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00056]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmUserDataTxPdu	
BSW Parameter		BSW Type
UdpNmTxUserDataPduId	EcucIntegerParamDef	
BSW Description		
This parameter defines the Handle ID of the NM User Data I-PDU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00058]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmUserDataTxPdu	
BSW Parameter		BSW Type
UdpNmTxUserDataPduRef	EcucReferenceDef	
BSW Description		
Reference to the NM User Data I-PDU in the global PDU collection.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00057]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		BSW Type

UdpNmWaitBusSleepTime	EcucFloatParamDef
BSW Description	
<p>Timeout for bus calm down phase. It denotes the time in [s] how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.</p> <p>It shall be equal for all nodes in the cluster. It shall be long enough to empty all Tx-buffer empty.</p>	
Template Description	
Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.	
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmWaitBusSleepTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00021]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	BSW Type
UdpNmComControlEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00013]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	BSW Type
UdpNmComUserDataSupport	EcucBooleanParamDef
BSW Description	
Enable/disable the user data support.	
Template Description	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping	
Mapping Rule	Mapping Type
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00055]

BSW Module		BSW Context	
UdpNm		UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type	
UdpNmCoordinatorSyncSupport		EcucBooleanParamDef	
BSW Description			
Enables/disables the coordinator synchronization support.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_UdpNm_00059]	

BSW Module		BSW Context	
UdpNm		UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type	
UdpNmDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_UdpNm_00002]	

BSW Module		BSW Context	
UdpNm		UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type	
UdpNmImmediateRestartEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling the immediate transmission of a NM PACKET upon bus-communication request in Prepare-Bus-Sleep mode. Must not be defined if UdpNmPassiveModeEnabled== true.			
Template Description			
Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.			
M2 Parameter			
SystemTemplate::NetworkManagement::UdpNmClusterCoupling.nmImmediateRestartEnabled			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_UdpNm_00009]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmNumberOfChannels		EcucIntegerParamDef
BSW Description		
Number of NM channels allowed within one ECU.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00014]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling support of the Passive Mode.		
Template Description		
Enables support of the Passive Mode. The passive mode is configurable per channel.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmPassiveModeEnabled		
Mapping Rule		Mapping Type
1:1 mapping nmNode.nmPassiveModeEnabled shall always have the same value in all Nm Clusters with the same bus protocol in the scope of one EcuInstance.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00010]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmPduRxIndicationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication.		
This parameter shall be derived from NmPduRxIndicationEnabled.		
Template Description		
Switch for enabling the PDU Rx Indication.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00011]

BSW Module	BSW Context	

UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmPnEiraCalcEnabled		EcucBooleanParamDef
BSW Description		
Specifies if UdpNm calculates the PN request information for internal and external requests. (EIRA) true: PN request are calculated false: PN request are not calculated		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00066]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmPnEiraRxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack. Only one SduRef is required for UdpNm because the EIRA is the aggregation over all Ethernet Channels.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00062]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmPnInfo		EcucParamConfContainerDef
BSW Description		
PN information configuration		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00067]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo	
BSW Parameter		BSW Type

UdpNmPnFilterMaskByte	EcucParamConfContainerDef
BSW Description	
PN information configuration	
Template Description	
Bit mask for Ethernet Payload used to configure the Ethernet Transceiver for partial network wakeup.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pncFilterDataMask	
Mapping Rule	Mapping Type
<p>For one EcuInstance all contributing EthernetCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the UdpNmPnFilterMaskByteIndex and UdpNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a UdpNmPnFilterMaskByte with UdpNmPnFilterMaskByteIndex = 5 and UdpNmPnFilterMaskByteValue = 128.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00070]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo/UdpNmPnFilterMaskByte
BSW Parameter	BSW Type
UdpNmPnFilterMaskByteIndex	EcucIntegerParamDef
BSW Description	
Index of the filter mask byte. Specifies the position within the filter mask byte array.	
Template Description	
Bit mask for Ethernet Payload used to configure the Ethernet Transceiver for partial network wakeup.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pncFilterDataMask	
Mapping Rule	Mapping Type
<p>For one EcuInstance all contributing EthernetCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the UdpNmPnFilterMaskByteIndex and UdpNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a UdpNmPnFilterMaskByte with UdpNmPnFilterMaskByteIndex = 5 and UdpNmPnFilterMaskByteValue = 128.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00071]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo/UdpNmPnFilterMaskByte
BSW Parameter	BSW Type

UdpNmPnFilterMaskByteValue	EcucIntegerParamDef
BSW Description	
Parameter to configure the filter mask byte.	
Template Description	
Bit mask for Ethernet Payload used to configure the Ethernet Transceiver for partial network wakeup.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationConnector.pncFilterDataMask	
Mapping Rule	Mapping Type
<p>For one EcuInstance all contributing EthernetCommunicationConnector.pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value.</p> <p>In order to get the UdpNmPnFilterMaskByteIndex and UdpNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way.</p> <p>E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2^{63} this will end up in a UdpNmPnFilterMaskByte with UdpNmPnFilterMaskByteIndex = 5 and UdpNmPnFilterMaskByteValue = 128.</p>	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00072]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo
BSW Parameter	BSW Type
UdpNmPnInfoLength	EcucIntegerParamDef
BSW Description	
Specifies the length of the PN request information in the NM message.	
Template Description	
Length of the partial networking request release information vector (in bytes).	
M2 Parameter	
SystemTemplate::System.pncVectorLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00069]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo
BSW Parameter	BSW Type
UdpNmPnInfoOffset	EcucIntegerParamDef
BSW Description	
Specifies the offset of the PN request information in the NM message.	
Template Description	
Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.	
M2 Parameter	
SystemTemplate::System.pncVectorOffset	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID

valid	[ECUC_UdpNm_00068]
-------	--------------------

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	BSW Type	
UdpNmPnResetTime	EcucFloatParamDef	
BSW Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA. The value shall be the same for every channel. Thus it is a global config parameter.		
Template Description		
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.pnResetTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00065]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	BSW Type	
UdpNmRemoteSleepIndEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling remote sleep indication support.		
This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmRemoteSleepIndEnabled.		
Template Description		
Switch for enabling remote sleep indication support.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_UdpNm_00005]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	BSW Type	
UdpNmStateChangeIndEnabled	EcucBooleanParamDef	
BSW Description		
Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NmStateChangeIndEnabled.		
Template Description		
Enables the CAN Network Management state change notification.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00012]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling user data support.		
This parameter shall be derived from NmUserDataEnabled.		
Template Description		
Switch for enabling user data support.		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00004]	

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		BSW Type
UdpNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_UdpNm_00003]	

C.6.11 SomelpTp

BSW Module	BSW Context	
SomelpTp	SomelpTp	
BSW Parameter		BSW Type
SomelpTpChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the SomelpTp channel.		
Template Description		
This element is used to assign properties to SomeipTpConnections that are referencing this SomeipTpChannel.		
M2 Parameter		
SystemTemplate::TransportProtocols::SomeipTpChannel		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SomelpTp_00003]	

BSW Module	BSW Context	
SomelpTp	SomelpTp/SomelpTpChannel	
BSW Parameter		BSW Type
SomelpTpNPduSeparationTime		EcucFloatParamDef
BSW Description		
Sets the duration of the minimum time in seconds the SomelpTp module shall wait between the transmissions of N-PDUs.		
Template Description		
Sets the duration of the minimum time in seconds the SOME/IP TP module shall wait between the transmissions of NPdus.		
M2 Parameter		
SystemTemplate::TransportProtocols::SomeipTpChannel.separationTime		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SomelpTp_00006]	

BSW Module	BSW Context	
SomelpTp	SomelpTp/SomelpTpChannel	
BSW Parameter		BSW Type
SomelpTpRxNSdu		EcucParamConfContainerDef
BSW Description		
The following parameters needs to be configured for each N-SDU which has to be passed as one assembled RxPdu to the upper layer.		
Template Description		
Reference to an IPdu that is segmented by the Transport Protocol.		
M2 Parameter		
SystemTemplate::TransportProtocols::SomeipTpConnection.tpSdu		
Mapping Rule		Mapping Type
Create reference if a SomeipTpConnection exists that points to the SomeipTpChannel that aggregates the SomelpTpRxNSdu and references a PduTriggering in the role tpSdu that is received by the EcuInstance that is contained in the Ecu Extract.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_SomelpTp_00008]	

BSW Module	BSW Context	
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpRxNSdu	
BSW Parameter		BSW Type
SomelpTpRxNPdu		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the NPdu that is received from a lower layer		
Template Description		
Reference to the segmented IPdu.		
M2 Parameter		

SystemTemplate::TransportProtocols::SomeipTpConnection.transportPdu	
Mapping Rule	Mapping Type
Create container if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is received by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00011]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpRxNSdu/SomelpTpRxNPdu
BSW Parameter	BSW Type
SomelpTpRxNPduHandleId	EcucIntegerParamDef
BSW Description	
This parameter defines the handle ID that is used by the PduR when calling SomelpTp_RxIndication.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00013]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpRxNSdu/SomelpTpRxNPdu
BSW Parameter	BSW Type
SomelpTpRxNPduRef	EcucReferenceDef
BSW Description	
Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
Template Description	
Reference to the segmented IPdu.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpConnection.transportPdu	
Mapping Rule	Mapping Type
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is received by the EcuInstance that is contained in the Ecu Extract.	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00012]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpRxNSdu
BSW Parameter	BSW Type
SomelpTpRxSduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack that represents the assembled RxPdu which is passed via the PduR to the upper layer.	
Template Description	
Reference to an IPdu that is segmented by the Transport Protocol.	
M2 Parameter	

SystemTemplate::TransportProtocols::SomeipTpConnection.tpSdu	
Mapping Rule	Mapping Type
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role tpSdu that is received by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomeipTp_00010]

BSW Module	BSW Context
SomeipTp	SomeipTp/SomeipTpChannel
BSW Parameter	BSW Type
SomeipTpRxTimeoutTime	EcucFloatParamDef
BSW Description	
Timer to monitor the successful reception. It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received. The value shall be calculated as follows: (SomeipTpRxTimeoutTime = SomeipTpNPduSeparationTime + budget), where the time budget compensates intermediary hops and jitters within the ECU implementation.	
Template Description	
Timer to monitor the successful reception. It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpChannel.rxTimeoutTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomeipTp_00023]

BSW Module	BSW Context
SomeipTp	SomeipTp/SomeipTpChannel
BSW Parameter	BSW Type
SomeipTpTxNSdu	EcucParamConfContainerDef
BSW Description	
The following parameters needs to be configured for each N-SDU that the SomeipTp module transmits via the SomeipTpChannel.	
Template Description	
Reference to an IPdu that is segmented by the Transport Protocol.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpConnection.tpSdu	
Mapping Rule	Mapping Type
Create reference if a SomeipTpConnection exists that points to the SomeipTp Channel that aggregates the SomeipTpTxNSdu and references a PduTriggering in the role tpSdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomeipTp_00009]

BSW Module	BSW Context
SomeipTp	SomeipTp/SomeipTpChannel/SomeipTpTxNSdu
BSW Parameter	BSW Type
SomeipTpTxNPdu	EcucParamConfContainerDef

BSW Description	
This container contains the configuration parameters of the segmented Tx NPdus that are transmitted to a lower layer.	
Template Description	
Reference to the segmented IPdu.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpConnection.transportPdu	
Mapping Rule	Mapping Type
Create container if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00016]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpTxNSdu/SomelpTpTxNPdu
BSW Parameter	BSW Type
SomelpTpTxNPduHandleId	EcuIntegerParamDef
BSW Description	
This parameter defines the handle ID that is used by PduR when calling SomelpTp_TriggerTransmit.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00017]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpTxNSdu/SomelpTpTxNPdu
BSW Parameter	BSW Type
SomelpTpTxNPduRef	EcuReferenceDef
BSW Description	
Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
Template Description	
Reference to the segmented IPdu.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpConnection.transportPdu	
Mapping Rule	Mapping Type
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00018]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpTxNSdu
BSW Parameter	BSW Type
SomelpTpTxNSduHandleId	EcuIntegerParamDef

BSW Description	
This parameter defines the handle ID of the NSdu that represents the original TxSdu which is segmented and passed via the PduR to the lower layer. This handle ID is used by PduR when calling SomelpTp_Transmit.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00020]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpChannel/SomelpTpTxNSdu
BSW Parameter	BSW Type
SomelpTpTxNSduRef	EcucReferenceDef
BSW Description	
Reference to a global Pdu in the COM-Stack that represents the original TxSdu which is segmented and passed via the PduR to the lower layer.	
Template Description	
Reference to an IPdu that is segmented by the Transport Protocol.	
M2 Parameter	
SystemTemplate::TransportProtocols::SomeipTpConnection.tpSdu	
Mapping Rule	Mapping Type
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role tpSdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00015]

BSW Module	BSW Context
SomelpTp	SomelpTp
BSW Parameter	BSW Type
SomelpTpGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the SomelpTp module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00002]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpGeneral
BSW Parameter	BSW Type
SomelpTpDevErrorDetect	EcucBooleanParamDef

BSW Description	
Switches the Development Error Detection and Notification ON or OFF.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00004]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpGeneral
BSW Parameter	BSW Type
SomelpTpRxMainFunctionPeriod	EcucFloatParamDef
BSW Description	
This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionRx.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00021]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpGeneral
BSW Parameter	BSW Type
SomelpTpTxMainFunctionPeriod	EcucFloatParamDef
BSW Description	
This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionTx.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00005]

BSW Module	BSW Context
SomelpTp	SomelpTp/SomelpTpGeneral
BSW Parameter	BSW Type
SomelpTpVersionInfoApi	EcucBooleanParamDef
BSW Description	

Activates the SomelpTp_GetVersionInfo() API. TRUE: Enables the SomelpTp_GetVersionInfo() API. FALSE: SomelpTp_GetVersionInfo() API is not included.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_SomelpTp_00019]

C.6.12 Wireless Ethernet Driver Mapping

BSW Module	BSW Context	
WEth	WEth	
BSW Parameter		BSW Type
WEthConfigSet		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR WEth module.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_WEth_00015]

BSW Module	BSW Context	
WEth	WEth/WEthConfigSet	
BSW Parameter		BSW Type
WEthCtrlConfig		EcucParamConfContainerDef
BSW Description		
Configuration of the individual controller		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_WEth_00006]

BSW Module	BSW Context	
WEth	WEth/WEthConfigSet/WEthCtrlConfig	
BSW Parameter		BSW Type
WEthCtrlEcucPartitionRef		EcucReferenceDef
BSW Description		

Maps the Wireless Ethernet controller to zero or one ECUC partitions. The ECUC partition referenced is a subset of the ECUC partitions where the Wireless Ethernet driver is mapped to.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00039]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthCtrlId	EcucIntegerParamDef
BSW Description	
Specifies the instance ID of the configured controller.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00007]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthCtrlPhyAddress	EcucStringParamDef
BSW Description	
Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order.	
Regular Expression: [0-9a-fA-F]{2}[:-][0-9a-fA-F]{2}{5}	
Template Description	
Media Access Control address (MAC address) that uniquely identifies each EthernetCommunicationController in the network.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationController.mac UnicastAddress	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00020]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthCtrlRxBufLenByte	EcucIntegerParamDef
BSW Description	

Limits the maximum receive buffer length (frame length) in bytes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00008]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthCtrlTxBufLenByte	EcucIntegerParamDef
BSW Description	
Limits the maximum transmit buffer length (frame length) in bytes.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00009]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthDemEventParameterRefs	EcucParamConfContainerDef
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00016]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig/WEthDemEventParameterRefs
BSW Parameter	BSW Type
WETH_E_ACCESS	EcucReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Controller access failed" has occurred.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00017]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthRxBufTotal	EcucIntegerParamDef
BSW Description	
Configures the number of receive buffers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00013]

BSW Module	BSW Context
WEth	WEth/WEthConfigSet/WEthCtrlConfig
BSW Parameter	BSW Type
WEthTxBufTotal	EcucIntegerParamDef
BSW Description	
Configures the number of transmit buffers.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00014]

BSW Module	BSW Context
WEth	WEth
BSW Parameter	BSW Type
WEthGeneral	EcucParamConfContainerDef
BSW Description	
General configuration of Wireless Ethernet Driver module.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type

	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00001]

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter		BSW Type
WEthDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Default Error Tracer (Det) detection and notification ON or OFF. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_WEth_00003]	

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter		BSW Type
WEthEcucPartitionRef		EcucReferenceDef
BSW Description		
Maps the Wireless Ethernet driver to zero or multiple ECUC partitions to make the modules API available in this partition. The Wireless Ethernet driver will operate as an independent instance in each of the partitions.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_WEth_00038]	

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter		BSW Type
WEthGetWEtherStatsApi		EcucBooleanParamDef
BSW Description		
Enables / Disables WEth_GetWEtherStats_32 and WEth_GetWEtherStats_64 API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_WEth_00036]

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter	BSW Type	
WEthIndex	EcucIntegerParamDef	
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_WEth_00018]	

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter	BSW Type	
WEthMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
Specifies the period of main function WEth_MainFunction in seconds. Wireless Ethernet driver does not require this information but the BSW scheduler.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_WEth_00022]	

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter	BSW Type	
WEthUpdatePhysAddrFilter	EcucBooleanParamDef	
BSW Description		
Enables/Disables optional API WEth_UpdatePhysAddrFilter.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_WEth_00019]	

BSW Module	BSW Context	
WEth	WEth/WEthGeneral	
BSW Parameter		BSW Type
WEthVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables / Disables version info API.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_WEth_00004]

C.7 Diagnostic

C.7.1 Dcm Mapping

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
BSW Parameter		BSW Type
DcmDslProtocolRxConnectionId		EcucIntegerParamDef
BSW Description		
Unique identifier of the tester which uses this connection for diagnostic communication.		
Template Description		
An ECU specific ID for responses of diagnostic routines.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstanceProps.diagnosticAddress		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Dcm_00826]

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
BSW Parameter		BSW Type
DcmDslProtocolRxTesterSourceAddr		EcucIntegerParamDef
BSW Description		
Tester source address uniquely describes a client and will be used e.g within the jump to Bootloader interfaces. This parameter is not required for generic connections (DcmPbus with MetaDataLength >= 1).		
Template Description		
An ECU specific ID for responses of diagnostic routines.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstanceProps.diagnosticAddress		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_01115]

C.8 Time management

C.8.1 StbM Time Management

BSW Module	BSW Context	
StbM	StbM	
BSW Parameter	BSW Type	
StbMGeneral	EcucParamConfContainerDef	
BSW Description	This container holds the general parameters of the Synchronized Time-base Manager	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00002]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMDevErrorDetect	EcucBooleanParamDef	
BSW Description	Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00012]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMEcucPartitionRef	EcucReferenceDef	
BSW Description	Reference to EcucPartition, where StbM module is assigned to.	
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00069]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter		BSW Type
StbMGetCurrentTimeExtendedAvailable		EcucBooleanParamDef
BSW Description		
This allows to define whether an additional variant of the API GetCurrentTime with a 64 bit argument is provided.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00032]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter		BSW Type
StbMGptTimerRef		EcucReferenceDef
BSW Description		
This represents an optional sub-container in case any Time Notification Customer is configured.		
The designated GPT timer has to be configured to have a tick duration of one micro second.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00039]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter		BSW Type
StbMMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Schedule period of the main function StbM_MainFunction. Unit: [s].		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00027]

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMTimeRecordingSupport	EcucBooleanParamDef	
BSW Description		
Enables/Disables the usage of the recording functionality for Synchronized and Offset timebases for Global Time precision measurement purpose.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00038]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMTimeValidationSupport	EcucBooleanParamDef	
BSW Description		
Enables/Disables the usage of the time recording functionality for Synchronized timebases for Time Validation purpose.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00070]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMTimerStartThreshold	EcucFloatParamDef	
BSW Description		
This interval defines, when a GPT Timer shall be started for Time Notification Customers for which the corresponding Customer Timer is running [unit: seconds].		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00063]	

BSW Module	BSW Context	
StbM	StbM/StbMGeneral	
BSW Parameter	BSW Type	
StbMVersionInfoApi	EcucBooleanParamDef	
BSW Description	Activate/Deactivate the version information API (StbM_GetVersionInfo). True: version information API activated False: version information API deactivated.	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00013]	

BSW Module	BSW Context	
StbM	StbM	
BSW Parameter	BSW Type	
StbMSynchronizedTimeBase	EcucParamConfContainerDef	
BSW Description	Synchronized time.base collects the information about a specific time-base provider within the system.	
Template Description	This represents the ability to define a global time domain.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeDomain	
Mapping Rule	Mapping Type	
For each GlobalTimeDomain where - the configured Ecu is connected to as slave or - the configured Ecu is connected to as master if the Ecu is not in the role of a GlobalTimeGateway for this GlobalTimeDomain an instance of StbMSynchronizedTimeBase shall be created.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00003]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter	BSW Type	
StbMAllowSystemWideGlobalTimeMaster	EcucBooleanParamDef	
BSW Description	For postbuild variant of the StbM this parameter has to be set to true for a Global Time Master that may act as a system-wide source of time. Otherwise no corresponding service ports/interfaces is provided. The Global Time Master functionality behind the service ports/interfaces has to be enabled/disabled separately via parameter StbMIsSystemWideGlobalTimeMaster.	
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00066]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type
StbMClearTimeleapCount		EcucIntegerParamDef
BSW Description		
This attribute describes the required number of updates to the Time Base where the time difference to the previous value has to remain below StbMTimeLeapPastThreshold/StbMTimeLeapFutureThreshold until the TIMELEAP_PAST/TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base is cleared.		
Template Description		
Defines the required number of updates to the Time Base where the time difference to the previous received value has to remain within the bounds of timeLeapFutureThreshold and timeLeapPastThreshold until that Time Base is considered healed.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave.timeLeapHealingCounter		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00037]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type
StbMIsSystemWideGlobalTimeMaster		EcucBooleanParamDef
BSW Description		
This parameter shall be set to true for a Global Time Master that acts as a system-wide source of time information with respect to Global Time.		
It is possible that several Global Time Masters exist that have set this parameter set to true because the Global Time Masters exist once per Global Time Domain and one ECU may own several Global Time Domains on different buses it is connected to.		
Template Description		
If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeMaster.isSystemWideGlobalTimeMaster		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00036]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type
StbMLocalTimeClock		EcucParamConfContainerDef
BSW Description		
References the hardware reference clock of this Synchronized Time Base.		

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00047]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMLocalTimeClock
BSW Parameter	BSW Type
StbMClockFrequency	EcucIntegerParamDef
BSW Description	
Represents the frequency [Hz] of the HW reference clock used by the StbM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00051]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMLocalTimeClock
BSW Parameter	BSW Type
StbMClockPrescaler	EcucIntegerParamDef
BSW Description	
Represents the prescaler to calculate the resulting frequency of the HW reference clock used by the StbM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00052]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMLocalTimeClock
BSW Parameter	BSW Type
StbMLocalTimeHardware	EcucChoiceReferenceDef
BSW Description	
Reference to the local time hardware.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00053]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type
StbMNotificationCustomer		EcucParamConfContainerDef
BSW Description		
This container holds the configuration of a notification customer, which is notified is informed about the occurrence of a Time-base related event.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00050]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMNotificationCustomer	
BSW Parameter		BSW Type
StbMNotificationCustomerId		EcucIntegerParamDef
BSW Description		
Identification of a event notification customer.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00062]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMNotificationCustomer	
BSW Parameter		BSW Type
StbMTimeNotificationCallback		EcucFunctionNameDef
BSW Description		
Name of the customer specific notification callback function, which shall be called, if the time previously set by the customer is reached.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	

valid	[ECUC_StbM_00064]
-------	-------------------

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter	BSW Type	
StbMNotificationInterface	EcucEnumerationParamDef	
BSW Description		
The parameter defines what type of interface shall be used to notify a customer of a status event.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00068]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter	BSW Type	
StbMOffsetTimeBase	EcucReferenceDef	
BSW Description		
This is the reference to the Synchronized Time-Base this Offset Time-Base is based on. This reference makes the containing StbMSynchronizedTimeBase an Offset Time-Base.		
Template Description		
Reference to a synchronized time domain this offset time domain is based on. The reference source is the offset time domain. The reference target is the synchronized time domain.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeDomain.offsetTimeDomain		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00030]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter	BSW Type	
StbMStatusNotificationCallback	EcucFunctionNameDef	
BSW Description		
Name of the customer specific status notification callback function, which shall be called, if a non-masked status event occurs.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00046]

BSW Module		BSW Context	
StbM		StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type	
StbMStatusNotificationMask		EcucIntegerParamDef	
BSW Description			
The parameter defines the initial value for NotificationMask mask, which defines the events for which the event notification callback function shall be called.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00045]	

BSW Module		BSW Context	
StbM		StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type	
StbMStoreTimebaseNonVolatile		EcucEnumerationParamDef	
BSW Description			
This allows for specifying that the Time Base shall be stored in the NvRam.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00031]	

BSW Module		BSW Context	
StbM		StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type	
StbMSyncLossTimeout		EcucFloatParamDef	
BSW Description			
This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain. Unit: seconds			
Template Description			
This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain.			
M2 Parameter			
SystemTemplate::GlobalTime::GlobalTimeDomain.syncLossTimeout			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00028]	

BSW Module		BSW Context	
StbM		StbM/StbMSynchronizedTimeBase	

BSW Parameter		BSW Type	
StbMSynchronizedTimeBaseIdentifier		EcucIntegerParamDef	
BSW Description			
<p>Identification of a Synchronized TimeBase via a unique identifier.</p> <p>Range:</p> <ul style="list-style-type: none"> * 0 .. 15: Synchronized Time Bases * 16 .. 31: Offset Time Bases * 32 .. 127: Pure Local Time Bases * 128 .. 65535: Reserved 			
Template Description			
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.			
M2 Parameter			
SystemTemplate::GlobalTime::GlobalTimeDomain.domainId			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00021]	

BSW Module	BSW Context		
StbM	StbM/StbMSynchronizedTimeBase		
BSW Parameter		BSW Type	
StbMTimeCorrection		EcucParamConfContainerDef	
BSW Description			
Collects the information relevant for the rate- and offset correction of a Time Base.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00048]	

BSW Module	BSW Context		
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection		
BSW Parameter		BSW Type	
StbMAllowMasterRateCorrection		EcucBooleanParamDef	
BSW Description			
<p>This attribute describes whether the rate correction value of a Time Base can be set by StbM_SetRateCorrection():</p> <ul style="list-style-type: none"> * false: the rate correction value can not be set by StbM_SetRateCorrection() * true: the rate correction value can be set by StbM_SetRateCorrection() 			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_StbM_00043]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection	
BSW Parameter	BSW Type	
StbMMasterRateDeviationMax	EcucIntegerParamDef	
BSW Description	This attribute describes the maximum allowed absolute value of the rate deviation value to be set by StbM_SetRateCorrection() [unit: ppm].	
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00044]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection	
BSW Parameter	BSW Type	
StbMOffsetCorrectionAdaptionInterval	EcucFloatParamDef	
BSW Description	Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation [unit: seconds].	
Template Description	Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeCorrectionProps.offsetCorrectionAdaptionInterval	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00057]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection	
BSW Parameter	BSW Type	
StbMOffsetCorrectionJumpThreshold	EcucFloatParamDef	
BSW Description	Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump [unit: seconds].	
Template Description	Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeCorrectionProps.offsetCorrectionJumpThreshold	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00056]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection	
BSW Parameter	BSW Type	
StbMRateCorrectionMeasurementDuration	EcucFloatParamDef	
BSW Description	Definition of the time span [s] which is used to calculate the rate deviation.	
Template Description	Definition of the time span which is used to calculate the rate deviation.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeCorrectionProps.rateCorrectionMeasurementDuration	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00054]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection	
BSW Parameter	BSW Type	
StbMRateCorrectionsPerMeasurementDuration	EcucIntegerParamDef	
BSW Description	Number of simultaneous rate measurements to determine the current rate deviation.	
Template Description	Defines the number of simultaneous rate measurements to determine the current rate deviation.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeCorrectionProps.rateCorrectionsPerMeasurementDuration	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00055]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter	BSW Type	
StbMTimeLeapFutureThreshold	EcucFloatParamDef	
BSW Description	This represents the maximum allowed positive difference between a newly received Global Time Base value and the current Local Time Base value [unit: seconds].	
Template Description	Defines the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value.	
M2 Parameter	SystemTemplate::GlobalTime::GlobalTimeSlave.timeLeapFutureThreshold	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00041]	

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	

BSW Parameter		BSW Type
StbMTimeLeapPastThreshold		EcucFloatParamDef
BSW Description		
This represents the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds].		
Template Description		
Defines the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave.timeLeapPastThreshold		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00042]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		BSW Type
StbMTimeRecording		EcucParamConfContainerDef
BSW Description		
Collects the information relevant for configuration of the precision measurement of a Time Base.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00049]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeRecording	
BSW Parameter		BSW Type
StbMEnableTimeValidation		EcucBooleanParamDef
BSW Description		
Enables/disables time recording for Time Validation for a specific Time Base.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_StbM_00071]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeRecording	
BSW Parameter		BSW Type
StbMOffsetTimeRecordBlockCallback		EcucFunctionNameDef
BSW Description		

Name of the customer specific callback function, which shall be called, if a measurement data for a Offset Time Base are available.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00061]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeRecording
BSW Parameter	BSW Type
StbMOffsetTimeRecordTableBlockCount	EcucIntegerParamDef
BSW Description	
Represents the number of Blocks used for queing time measurement events for the Offset Time Base Record Table.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00059]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeRecording
BSW Parameter	BSW Type
StbMSyncTimeRecordBlockCallback	EcucFunctionNameDef
BSW Description	
Name of the customer specific callback function, which shall be called, if a measurement data for a Synchronized Time Base are available.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00060]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeRecording
BSW Parameter	BSW Type
StbMSyncTimeRecordTableBlockCount	EcucIntegerParamDef
BSW Description	
Represents the number of Blocks used for queing time measurement events for the Synchronized Time Base Record Table.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00058]

BSW Module	BSW Context
StbM	StbM
BSW Parameter	BSW Type
StbMTriggeredCustomer	EcucParamConfContainerDef
BSW Description	
The triggered customer is directly triggered by the Synchronized Time-base Manager by getting synchronized with the current (global) definition of time and passage of time.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00004]

BSW Module	BSW Context
StbM	StbM/StbMTriggeredCustomer
BSW Parameter	BSW Type
StbMOSScheduleTableRef	EcucReferenceDef
BSW Description	
Mandatory reference to synchronized OS ScheduleTable, which will be explicitly synchronized by the StbM.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00007]

BSW Module	BSW Context
StbM	StbM/StbMTriggeredCustomer
BSW Parameter	BSW Type
StbMSynchronizedTimeBaseRef	EcucReferenceDef
BSW Description	
Mandatory reference to the required synchronized time-base.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00010]	

BSW Module	BSW Context	
StbM	StbM/StbMTriggeredCustomer	
BSW Parameter		BSW Type
StbMTriggeredCustomerPeriod		EcucIntegerParamDef
BSW Description		
The triggering period of the triggered customer, called by the StbM_MainFunction.		
The period is documented in microseconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_StbM_00020]	

C.8.2 CAN Time Management

BSW Module	BSW Context	
CanTSyn	CanTSyn	
BSW Parameter		BSW Type
CanTSynGeneral		EcucParamConfContainerDef
BSW Description		
This container holds the general parameters of the CAN-specific Synchronized Time-base Manager		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00003]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGeneral	
BSW Parameter		BSW Type
CanTSynDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the development error detection and notification on or off.		
* true: detection and notification is enabled.		
* false: detection and notification is disabled.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00002]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGeneral
BSW Parameter	BSW Type
CanTSynMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Schedule period of the main function CanTSyn_MainFunction. Unit: [s].	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00019]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGeneral
BSW Parameter	BSW Type
CanTSynTimeValidationSupport	EcucBooleanParamDef
BSW Description	
Switches support for Time Validation on or off.	
<ul style="list-style-type: none"> - true: Time Validation is enabled. - false: Time Validation is disabled 	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00050]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGeneral
BSW Parameter	BSW Type
CanTSynVersionInfoApi	EcucBooleanParamDef
BSW Description	
Activate/Deactivate the version information API (CanTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00023]	

BSW Module	BSW Context	
CanTSyn	CanTSyn	
BSW Parameter	BSW Type	
CanTSynGlobalTimeDomain	EcucParamConfContainerDef	
BSW Description		
This represents the existence of a global time domain on CAN. The CanTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains.		
If the CanTSyn exists it is assumed that at least one global time domain exists.		
Template Description		
This represents the ability to define a global time domain.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeDomain		
Mapping Rule		Mapping Type
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the Ecuc Instance for which the ECU configuration is created.		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00004]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter	BSW Type	
CanTSynEnableTimeValidation	EcucBooleanParamDef	
BSW Description		
Enables/disables time recording for Time Validation for a specific Time Domain.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00051]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter	BSW Type	
CanTSynGlobalTimeDomainId	EcucIntegerParamDef	
BSW Description		
The global time domain ID.		
Template Description		
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.		

M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.domainId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00005]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	BSW Type
CanTSynGlobalTimeFupDataIDList	EcucParamConfContainerDef
BSW Description	
The DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00025]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeFupDataIDList
BSW Parameter	BSW Type
CanTSynGlobalTimeFupDataIDListElement	EcucParamConfContainerDef
BSW Description	
Element of the DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	Mapping Type
Value shall be derived from element of the ordered fupDataIDList.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00031]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeFupDataIDList/ CanTSynGlobalTimeFupDataIDListElement
BSW Parameter	BSW Type
CanTSynGlobalTimeFupDataIDListIndex	EcucIntegerParamDef
BSW Description	
Index of the DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00032]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeFupDataIDList/CanTSynGlobalTimeFupDataIDListElement	
BSW Parameter		BSW Type
CanTSynGlobalTimeFupDataIDListValue		EcucIntegerParamDef
BSW Description		
Value of the DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00033]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type
CanTSynGlobalTimeMaster		EcucParamConfContainerDef
BSW Description		
Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.		
Template Description		
This represents the generic concept of a global time master.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeMaster		
Mapping Rule		Mapping Type
The existence of the CanTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00007]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter		BSW Type
CanTSynCyclicMsgResumeTime		EcucFloatParamDef
BSW Description		
Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds		
Template Description		
Defines the minimum time between an "immediate" message and the next periodic message.		

M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster.immediateResumeTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00044]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	BSW Type
CanTSynGlobalTimeDebounceTime	EcucFloatParamDef
BSW Description	
This represents the configuration of a TX debounce time for SYNC, FUP, OFS and OFNS messages compared to a message before with the same PDU. Unit: seconds	
Template Description	
Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.debounceTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00045]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	BSW Type
CanTSynGlobalTimeMasterPdu	EcucParamConfContainerDef
BSW Description	
This container encloses the configuration of the PDU that is supposed to contain the global time information.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00009]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeMasterPdu
BSW Parameter	BSW Type
CanTSynGlobalTimeMasterConfirmationHandleId	EcucIntegerParamDef
BSW Description	
This represents the handle ID of the PDU that contains the global time information.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00008]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeMasterPdu	
BSW Parameter	BSW Type	
CanTSynGlobalTimePduRef	EcucReferenceDef	
BSW Description		
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.		
Template Description		
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeDomain.pduTriggering		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
CanTSynGlobalTimeTx_crcSecured	EcucEnumerationParamDef	
BSW Description		
This represents the configuration of whether or not CRC is supported.		
Template Description		
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::GlobalTimeCanMaster.crcSecured		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00015]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTx_crcSecured	
BSW Parameter	BSW Type	
CRC_NOT_SUPPORTED	EcucEnumerationLiteralDef	
BSW Description		
This represents a configuration where CRC is not supported.		
Template Description		
This indicates that CRC is not supported		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcNotSupported		

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTx_crcSecured	
BSW Parameter	BSW Type	
CRC_SUPPORTED	EcucEnumerationLiteralDef	
BSW Description		
This represents a configuration where CRC is supported.		
Template Description		
This indicates that CRC is supported		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcSupported		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
CanTSynGlobalTimeTxPeriod	EcucFloatParamDef	
BSW Description		
This represents configuration of the TX period. Unit: seconds		
Template Description		
This represents the period. Unit: seconds		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00017]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
CanTSynImmediateTimeSync	EcucBooleanParamDef	
BSW Description		
Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within CanTSyn_MainFunction().		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	

valid	[ECUC_CanTSyn_00043]
-------	----------------------

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
CanTSynMasterConfirmationTimeout	EcucFloatParamDef	
BSW Description		
This represents the confirmation timeout after transmission of each Timesync message.		
Unit: seconds.		
Template Description		
This represents the value for the confirmation timeout. Unit: seconds.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::GlobalTimeCanMaster.syncConfirmationTimeout		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00020]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
CanTSynTxTmacCalculated	EcucEnumerationParamDef	
BSW Description		
This parameter controls whether or not TMAC calculation shall be supported.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00047]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter	BSW Type	
CanTSynGlobalTimeOfnsDataIDList	EcucParamConfContainerDef	
BSW Description		
The DataIDList for OFNS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
The DataIDList for OFNS messages to calculate CRC.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.ofnsDataIDList		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00041]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfnsDataIDList	
BSW Parameter	BSW Type	
CanTSynGlobalTimeOfnsDataIDListElement	EcucParamConfContainerDef	
BSW Description		
Element of the DataIDList for OFNS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
The DataIDList for OFNS messages to calculate CRC.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.ofnsDataIDList		
Mapping Rule	Mapping Type	
Value shall be derived from element of the ordered ofnsDataIDList.	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00037]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfnsDataIDList/CanTSynGlobalTimeOfnsDataIDListElement	
BSW Parameter	BSW Type	
CanTSynGlobalTimeOfnsDataIDListIndex	EcucIntegerParamDef	
BSW Description		
Index of the DataIDList for OFNS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00038]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfnsDataIDList/CanTSynGlobalTimeOfnsDataIDListElement	
BSW Parameter	BSW Type	
CanTSynGlobalTimeOfnsDataIDListValue	EcucIntegerParamDef	
BSW Description		
Value of the DataIDList for OFNS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
	local	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00039]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type	
CanTSynGlobalTimeOfsDataIDList		EcucParamConfContainerDef	
BSW Description			
The DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
The DataIDList for OFS messages to calculate CRC.			
M2 Parameter			
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.ofsDataIDList			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00026]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfsDataIDList	
BSW Parameter		BSW Type	
CanTSynGlobalTimeOfsDataIDListElement		EcucParamConfContainerDef	
BSW Description			
Element of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
The DataIDList for OFS messages to calculate CRC.			
M2 Parameter			
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.ofsDataIDList			
Mapping Rule		Mapping Type	
Value shall be derived from element of the ordered ofsDataIDList.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00034]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfsDataIDList/ CanTSynGlobalTimeOfsDataIDListElement	
BSW Parameter		BSW Type	
CanTSynGlobalTimeOfsDataIDListIndex		EcucIntegerParamDef	
BSW Description			
Index of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00035]	

BSW Module		BSW Context	
------------	--	-------------	--

CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeOfsDataIDList/CanTSynGlobalTimeOfsDataIDListElement	
BSW Parameter		BSW Type
CanTSynGlobalTimeOfsDataIDListValue		EcucIntegerParamDef
BSW Description		
Value of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00036]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type
CanTSynGlobalTimeSecureTmacLength		EcucIntegerParamDef
BSW Description		
Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00046]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type
CanTSynGlobalTimeSlave		EcucParamConfContainerDef
BSW Description		
Configuration of a global time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.		
Template Description		
This represents the generic concept of a global time slave.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave		
Mapping Rule		Mapping Type
The existence of the CanTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn is aggregated at the EcucInstance for which the ECU configuration exists,		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00012]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
CanTSynGlobalTimeFollowUpTimeout	EcucFloatParamDef	
BSW Description		
Rx timeout for the follow-up message. This is only relevant for selected bus systems Unit:seconds		
Template Description		
Rx timeout for the follow-up message.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave.followUpTimeoutValue		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00006]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
CanTSynGlobalTimeMinMsgGap	EcucFloatParamDef	
BSW Description		
This parameter represents the configuration of a minimum message gap time for received Timesync messages compared to a message before with the same PDU. If PDUs are received more often in between than this parameter allows, they shall be ignored.		
Unit: seconds		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00049]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
CanTSynGlobalTimeSequenceCounterJumpWidth	EcucIntegerParamDef	
BSW Description		
The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC resp. two OFS messages.		
Template Description		
Specifies the maximum allowed gap of the sequence counter between two SYNC resp. two OFS messages.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::GlobalTimeCanSlave.sequenceCounterJumpWidth		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00011]

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter		BSW Type	
CanTSynGlobalTimeSlavePdu		EcucParamConfContainerDef	
BSW Description			
This container encloses the configuration of the PDU that is supposed to contain the global time information.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00014]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeSlavePdu	
BSW Parameter		BSW Type	
CanTSynGlobalTimePduRef		EcucReferenceDef	
BSW Description			
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.			
Template Description			
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves.			
M2 Parameter			
SystemTemplate::GlobalTime::GlobalTimeDomain.pduTriggering			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeSlavePdu	
BSW Parameter		BSW Type	
CanTSynGlobalTimeSlaveHandleId		EcucIntegerParamDef	
BSW Description			
This represents the handle ID of the PDU that contains the global time information.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00013]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
CanTSynRxCrcValidated	EcucEnumerationParamDef	
BSW Description		
Definition of whether or not validation of the CRC is supported.		
Template Description		
Definition of whether or not validation of the CRC is supported.		
M2 Parameter		
SystemTemplate::GlobalTime::CAN::GlobalTimeCanSlave.crcValidated		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_CanTSyn_00021]	

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated	
BSW Parameter	BSW Type	
CRC_IGNORED	EcucEnumerationLiteralDef	
BSW Description		
The Timesync module accepts Time Synchronization messages, which are CRC secured (without actually validating the CRC) and those, which are not CRC secured. That means, the Timesync module ignores the CRC.		
Template Description		
The CRC is supposed to be ignored		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcIgnored		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated	
BSW Parameter	BSW Type	
CRC_NOT_VALIDATED	EcucEnumerationLiteralDef	
BSW Description		
The Timesync module accepts only Time Synchronization messages, which are not CRC secured. All other Time Synchronization messages are ignored.		
Template Description		
The CRC is not supposed to be present. If CRC is present the message is ignored.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcNotValidated		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRx CrcValidated	
BSW Parameter		BSW Type
CRC_OPTIONAL		EcucEnumerationLiteralDef
BSW Description		
The Timesync module accepts only Time Synchronization messages which are not CRC secured and Time Synchronization messages which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.		
Template Description		
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcOptional		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRx CrcValidated	
BSW Parameter		BSW Type
CRC_VALIDATED		EcucEnumerationLiteralDef
BSW Description		
The Timesync module accepts only Time Synchronization messages, which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.		
Template Description		
This CRC is supposed to be validated.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcValidated		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave	
BSW Parameter		BSW Type
CanTSynRxTmacValidated		EcucEnumerationParamDef
BSW Description		
This parameter controls whether or not TMAC validation shall be supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00048]

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type	
CanTSynGlobalTimeSyncDataIDList		EcucParamConfContainerDef	
BSW Description			
The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
The DataIDList for SYNC messages to calculate CRC.			
M2 Parameter			
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.syncDataIDList			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00024]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSyncDataIDList	
BSW Parameter		BSW Type	
CanTSynGlobalTimeSyncDataIDListElement		EcucParamConfContainerDef	
BSW Description			
Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
The DataIDList for SYNC messages to calculate CRC.			
M2 Parameter			
SystemTemplate::GlobalTime::CAN::CanGlobalTimeDomainProps.syncDataIDList			
Mapping Rule		Mapping Type	
Value shall be derived from element of the ordered syncDataIDList.		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00028]	

BSW Module		BSW Context	
CanTSyn		CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSyncDataIDList/ CanTSynGlobalTimeSyncDataIDListElement	
BSW Parameter		BSW Type	
CanTSynGlobalTimeSyncDataIDListIndex		EcucIntegerParamDef	
BSW Description			
Index for the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.			
Template Description			
M2 Parameter			
Mapping Rule		Mapping Type	
		local	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_CanTSyn_00029]	

BSW Module		BSW Context	
------------	--	-------------	--

CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSyncDataIDList/CanTSynGlobalTimeSyncDataIDListElement	
BSW Parameter		BSW Type
CanTSynGlobalTimeSyncDataIDListValue		EcucIntegerParamDef
BSW Description		
Value of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00030]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type
CanTSynSynchronizedTimeBaseRef		EcucReferenceDef
BSW Description		
Mandatory reference to the required synchronized time-base.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00022]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		BSW Type
CanTSynUseExtendedMsgFormat		EcucBooleanParamDef
BSW Description		
Switches support for 16 Byte Timesync messages on or off (for CAN FD only)		
* true: CAN FD support is active: use at least 16 byte for Timesync messages (depending on configuration)		
* false: Classic CAN support is active: use always 8 byte for Timesync messages		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_CanTSyn_00042]

C.8.3 Ethernet Time Management

BSW Module		BSW Context	
EthTSyn		EthTSyn	
BSW Parameter		BSW Type	
EthTSynGeneral		EcucParamConfContainerDef	
BSW Description			
This container holds the general parameters of the Ethernet-specific Synchronized Time-base Manager			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00003]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type	
EthTSynDestPhyAddr		EcucStringParamDef	
BSW Description			
Destination Physical Address (MAC-Address).			
Destination Physical Hardware Address (MAC-Address) of EthTSyn-gPTP Frames. Input format has to match xx:xx:xx:xx:xx:xx, where x stands for a hex value between 0 and F.			
Template Description			
Defines the MAC multicast address the Ethernet time sync messages are communicated on.			
M2 Parameter			
SystemTemplate::GlobalTime::ETH::EthGlobalTimeDomainProps.destinationPhysicalAddress			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00058]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type	
EthTSynDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
			local
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00002]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type	
EthTSynEthIfFrameType		EcucReferenceDef	
BSW Description			
The chosen frame owner determines which frames (in respect to ethertype) are received.			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00062]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type	
EthTSynGlobalTimeRxToUplinkSwitchResidenceTime		EcucFloatParamDef	
BSW Description			
This parameter is specifying the default value used for the residence time of the Ethernet Switch [Ingress to Uplink].			
This value is used by the EthTSyn if the calculation of the residence time failed.			
Unit: seconds			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00060]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type	
EthTSynGlobalTimeUplinkToTxSwitchResidenceTime		EcucFloatParamDef	
BSW Description			
This parameter is specifying the default value used for the residence time of the Ethernet Switch [Uplink to Egress].			
This value is used by the EthTSyn if the calculation of the residence time failed.			
Unit: seconds			
Template Description			
M2 Parameter			
Mapping Rule			Mapping Type

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00061]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter	BSW Type	
EthTSynHardwareTimestampSupport	EcucBooleanParamDef	
BSW Description		
Activate/Deactivate the hardware time stamping functionality of the Ethernet hardware. True: Timestamp is retrieved from the Ethernet hardware False: Timestamp is retrieved from the StbM		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00018]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter	BSW Type	
EthTSynMainFunctionPeriod	EcucFloatParamDef	
BSW Description		
Schedule period of the main function EthTSyn_MainFunction. Unit: seconds.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00012]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter	BSW Type	
EthTSynMasterSlaveConflictDetection	EcucBooleanParamDef	
BSW Description		
Enables master / slave conflict detection and notification. * true: detection and notification is enabled. * false: detection and notification is disabled.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00075]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type
EthTSynMessageCompliance		EcucBooleanParamDef
BSW Description		
* true: IEEE 802.1AS compliant message format will be used. * false: IEEE 802.1AS message format with AUTOSAR extension will be used.		
Template Description		
Defines the compliance of the Ethernet time sync messages to specific standards.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthGlobalTimeDomainProps.messageCompliance		
Mapping Rule		Mapping Type
If EthGlobalTimeDomainProps.messageFormat = IEEE802_1AS then EthTSyn MessageCompliance shall be true. If EthGlobalTimeDomainProps.messageFormat = IEEE802_1AS_AUTOSAR then EthTSynMessageCompliance shall be false.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthTSyn_00029]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type
EthTSynSwitchMgmtRxMessageBufferCount		EcucIntegerParamDef
BSW Description		
This parameter is used to determine the amount of Rx message buffers available in the EthTSyn when EthTSyn is used in a Bridge configuration.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthTSyn_00059]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGeneral	
BSW Parameter		BSW Type
EthTSynTimeValidationSupport		EcucBooleanParamDef
BSW Description		
Switches support for time validation on or off.		
- true: time validation is enabled. - false: time validation is disabled.		

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00081]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGeneral
BSW Parameter	BSW Type
EthTSynVersionInfoApi	EcucBooleanParamDef
BSW Description	
Activate/Deactivate the version information API (EthTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00015]

BSW Module	BSW Context
EthTSyn	EthTSyn
BSW Parameter	BSW Type
EthTSynGlobalTimeDomain	EcucParamConfContainerDef
BSW Description	
This represents the existence of a global time domain on Ethernet. The EthTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains.	
If the EthTSyn exists it is assumed that at least one global time domain exists.	
Template Description	
This represents the ability to define a global time domain.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain	
Mapping Rule	Mapping Type
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the Ecuc Instance for which the ECU configuration is created.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00004]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynEnableTimeValidation	EcucBooleanParamDef

BSW Description	
Enables/disables time recording for time validation for a specific Time Domain.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00082]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynGlobalTimeDomainId	EcucIntegerParamDef
BSW Description	
The global time domain ID.	
Template Description	
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.domainId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00005]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynGlobalTimeFollowUpDataIDList	EcucParamConfContainerDef
BSW Description	
The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00030]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynGlobalTimeFollowUpDataIDList
BSW Parameter	BSW Type
EthTSynGlobalTimeFollowUpDataIDListElement	EcucParamConfContainerDef
BSW Description	
Element of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	

Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	Mapping Type
Value shall be derived from element of the ordered fupDataIDList.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00031]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynGlobalTimeFollowUpDataIDList/ EthTSynGlobalTimeFollowUpDataIDListElement
BSW Parameter	BSW Type
EthTSynGlobalTimeFollowUpDataIDListIndex	EcucIntegerParamDef
BSW Description	
Index of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00032]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynGlobalTimeFollowUpDataIDList/ EthTSynGlobalTimeFollowUpDataIDListElement
BSW Parameter	BSW Type
EthTSynGlobalTimeFollowUpDataIDListValue	EcucIntegerParamDef
BSW Description	
Value of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00033]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynGlobalTimeSecureTmacLength	EcucIntegerParamDef
BSW Description	
Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00077]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynPortConfig	EcucParamConfContainerDef
BSW Description	
Configuration of the EthTSyn-Ports within the TimeDomain.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00063]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynFramePrio	EcucIntegerParamDef
BSW Description	
This optional parameter, if present, indicates the priority of outgoing EthTSyn messages, if sent via VLAN (used for the 3-bit PCP field of the VLAN tag). If this optional parameter is not present, frames are sent without a priority and VLAN field.	
Template Description	
Defines which VLAN priority shall be assigned to a time sync message in case the message is sent using a VLAN tag.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeDomainProps.vlanPriority	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00034]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynGlobalTimeDebounceTime	EcucFloatParamDef
BSW Description	
This represents the configuration of a TX debounce time for Sync and Follow_Up messages compared to a message before with the same PDU. Unit: seconds	
Template Description	

Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.debounceTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00048]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynGlobalTimeEthIfRef	EcucReferenceDef
BSW Description	
This represents the reference to the Ethernet interface taken to fetch the global time information.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00065]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynGlobalTimeMinMsgGap	EcucFloatParamDef
BSW Description	
This parameter represents the configuration of a minimum message gap time for received Timesync messages compared to a message before with the same PDU. If PDUs are received more often in between than this parameter allows, they shall be ignored.	
Unit: seconds	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00078]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynPdelayConfig	EcucParamConfContainerDef
BSW Description	
Configuration of cyclic propagation delay measurement.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00068]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelayConfig
BSW Parameter	BSW Type
EthTSynGlobalTimePdelayRespEnable	EcucBooleanParamDef
BSW Description	
<p>This parameter allows disabling Pdelay_Resp / Pdelay_Resp_Follow_Up transmission, if no Pdelay_Req messages are expected.</p> <p>FALSE: No Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is disabled.</p> <p>TRUE: Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is enabled.</p>	
Template Description	
<p>Defines whether PDELAY RESPONSE and PDELAY RESPONSE FOLLOW UP shall be sent on this CouplingPort.</p>	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeManagedCouplingPort.pdelayResponseEnabled	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00069]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelayConfig
BSW Parameter	BSW Type
EthTSynGlobalTimePropagationDelay	EcucFloatParamDef
BSW Description	
<p>If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available.</p> <p>If cyclic propagation delay measurement is disabled, this parameter replaces a measured propagation delay by a fixed value.</p> <p>Unit: seconds</p>	
Template Description	
<p>If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available.</p> <p>If cyclic propagation delay measurement is disabled, this parameter defines a fixed value for the propagation delay.</p>	
M2 Parameter	

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::GlobalTimeCouplingPort Props.propagationDelay	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00070]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelay Config
BSW Parameter	BSW Type
EthTSynGlobalTimeTxPdelayReqPeriod	EcucFloatParamDef
BSW Description	
This represents configuration of the TX period for Pdelay_Req messages. A value of 0 disables the cyclic Pdelay measurement. Unit: seconds	
Template Description	
Defines the period for the pdelay request messages.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00071]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelay Config
BSW Parameter	BSW Type
EthTSynPdelayLatencyThreshold	EcucFloatParamDef
BSW Description	
Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, this value is discarded. Unit: seconds	
Template Description	
Threshold for calculated Pdelay. If a measured Pdelay exceeds pdelayLatencyThreshold, the mea- sured Pdelay value is discarded.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeManagedCouplingPort.pdelayLatencyThreshold	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00076]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPdelay Config
BSW Parameter	BSW Type

EthTSynPdelayRespAndRespFollowUpTimeout	EcucFloatParamDef
BSW Description	
Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received.	
A value of 0 deactivates this timeout observation.	
Unit: seconds	
Template Description	
Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 or not defining this attribute deactivates this timeout observation.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeManagedCouplingPort.pdelayRespAndRespFollowUpTimeout	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00074]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynPortRole	EcucChoiceContainerDef
BSW Description	
Specifying the Role of the EthTSyn-Port (Master or Slave).	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00067]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole
BSW Parameter	BSW Type
EthTSynGlobalTimeMaster	EcucParamConfContainerDef
BSW Description	
Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.	
Template Description	
This represents the generic concept of a global time master.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster	
Mapping Rule	Mapping Type
The existence of a EthTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn is aggregated at the EcuInstance for which the ECU configuration exists.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00008]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynCrcTimeFlagsTxSecured		EcucParamConfContainerDef
BSW Description		
This container collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthTSyn_00057]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	
BSW Parameter		BSW Type
EthTSynCrcCorrectionField		EcucBooleanParamDef
BSW Description		
correctionField from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcCorrectionField		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	
BSW Parameter		BSW Type
EthTSynCrcDomainNumber		EcucBooleanParamDef
BSW Description		
domainNumber from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcDomainNumber		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context
------------	-------------

EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	
BSW Parameter		BSW Type
EthTSynCrcMessageLength	EcucBooleanParamDef	
BSW Description		
messageLength from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
MessageLength from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcMessageLength		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	
BSW Parameter		BSW Type
EthTSynCrcPreciseOriginTimestamp	EcucBooleanParamDef	
BSW Description		
preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
Template Description		
PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcPreciseOriginTimestamp		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	
BSW Parameter		BSW Type
EthTSynCrcSequenceld	EcucBooleanParamDef	
BSW Description		
sequenceld from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
Sequenceld from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcSequenceld		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynCrcTimeFlagsTxSecured	

BSW Parameter		BSW Type	
EthTSynCrcSourcePortIdentity		EcucBooleanParamDef	
BSW Description			
sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.			
Template Description			
SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.			
M2 Parameter			
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcSourcePortIdentity			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type	
EthTSynCyclicMsgResumeTime		EcucFloatParamDef	
BSW Description			
Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds			
Template Description			
Defines the minimum time between an "immediate" message and the next periodic message.			
M2 Parameter			
SystemTemplate::GlobalTime::GlobalTimeMaster.immediateResumeTime			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00047]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type	
EthTSynGlobalTimeTxCrcSecured		EcucEnumerationParamDef	
BSW Description			
This represents the configuration of whether or not CRC is supported.			
Template Description			
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.			
M2 Parameter			
SystemTemplate::GlobalTime::ETH::GlobalTimeEthMaster.crcSecured			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_EthTSyn_00039]	

BSW Module		BSW Context	
EthTSyn		EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynGlobalTimeTxCrcSecured	
BSW Parameter		BSW Type	

CRC_NOT_SUPPORTED	EcucEnumerationLiteralDef
BSW Description	
This represents a configuration where CRC is not supported.	
Template Description	
This indicates that CRC is not supported	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcNotSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster/EthTSynGlobalTimeTxCrcSecured
BSW Parameter	BSW Type
CRC_SUPPORTED	EcucEnumerationLiteralDef
BSW Description	
This represents a configuration where CRC is supported.	
Template Description	
This indicates that CRC is supported	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster
BSW Parameter	BSW Type
EthTSynGlobalTimeTxPeriod	EcucFloatParamDef
BSW Description	
This represents configuration of the TX period. Unit: seconds	
Template Description	
This represents the period. Unit: seconds	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00010]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster
BSW Parameter	BSW Type
EthTSynImmediateTimeSync	EcucBooleanParamDef
BSW Description	

Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within EthTSyn_MainFunction().	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00046]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynTLVFollowUpOFSSubTLV		EcucBooleanParamDef
BSW Description		
This represents the configuration of whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used or not. * true: This represents a configuration where an AUTOSAR Follow_Up TLV OFS Sub-TLV is used. * false: This represents a configuration where an AUTOSAR Follow_Up TLV OFS Sub-TLV is not used.		
Template Description		
Defines whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynSubTlvConfig.ofsSubTlv		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00038]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynTLVFollowUpStatusSubTLV		EcucBooleanParamDef
BSW Description		
This represents the configuration of whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used or not. * true: This represents a configuration where an AUTOSAR Follow_Up TLV Status Sub-TLV is used. * false: This represents a configuration where an AUTOSAR Follow_Up TLV Status Sub-TLV is not used.		
Template Description		
Defines whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynSubTlvConfig.statusSubTlv		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00036]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynTLVFollowUpTimeSubTLV		EcucBooleanParamDef
BSW Description		
<p>This represents the configuration of whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used or not.</p> <p>* true: This represents a configuration where an AUTOSAR Follow_Up TLV Time Sub-TLV is used.</p> <p>* false: This represents a configuration where an AUTOSAR Follow_Up TLV Time Sub-TLV is not used.</p>		
Template Description		
Defines whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynSubTlvConfig.timeSubTlv		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthTSyn_00035]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynTLVFollowUpUserDataSubTLV		EcucBooleanParamDef
BSW Description		
<p>This represents the configuration of whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used or not.</p> <p>* true: This represents a configuration where an AUTOSAR Follow_Up TLV UserData Sub-TLV is used.</p> <p>* false: This represents a configuration where an AUTOSAR Follow_Up TLV UserData Sub-TLV is not used.</p>		
Template Description		
Defines whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynSubTlvConfig.userDataSubTlv		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EthTSyn_00037]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeMaster	
BSW Parameter		BSW Type
EthTSynTxTmacCalculated		EcucEnumerationParamDef
BSW Description		
This parameter controls whether or not TMAC calculation shall be supported.		
Template Description		
M2 Parameter		

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00079]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole	
BSW Parameter		BSW Type
EthTSynGlobalTimeSlave		EcucParamConfContainerDef
BSW Description		
Configuration of a time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.		
Template Description		
This represents the generic concept of a global time slave.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave		
Mapping Rule		Mapping Type
The existence of the EthTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00009]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave	
BSW Parameter		BSW Type
EthTSynCrcFlagsRxValidated		EcucParamConfContainerDef
BSW Description		
This container collects definitions which parts of the Follow_Up message elements shall be included in CRC validation.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EthTSyn_00050]	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated	
BSW Parameter		BSW Type
EthTSynCrcCorrectionField		EcucBooleanParamDef
BSW Description		
correctionField from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		

SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcCorrectionField	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated
BSW Parameter	BSW Type
EthTSynCrcDomainNumber	EcucBooleanParamDef
BSW Description	
domainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcDomainNumber	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated
BSW Parameter	BSW Type
EthTSynCrcMessageLength	EcucBooleanParamDef
BSW Description	
messageLength from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
MessageLength from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcMessageLength	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated
BSW Parameter	BSW Type
EthTSynCrcPreciseOriginTimestamp	EcucBooleanParamDef
BSW Description	
preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.	
Template Description	
PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcPreciseOriginTimestamp	
Mapping Rule	Mapping Type

1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated	
BSW Parameter	BSW Type	
EthTSynCrcSequenceld	EcucBooleanParamDef	
BSW Description		
sequenceld from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
Sequenceld from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcSequenceld		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynCrcFlagsRxValidated	
BSW Parameter	BSW Type	
EthTSynCrcSourcePortIdentity	EcucBooleanParamDef	
BSW Description		
sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
SystemTemplate::GlobalTime::ETH::EthTSynCrcFlags.crcSourcePortIdentity		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
EthTSynGlobalTimeFollowUpTimeout	EcucFloatParamDef	
BSW Description		
Timeout value of the Follow_Up message (of the subsequent Sync message).		
A value of 0 deactivates this timeout observation.		
Unit: seconds		
Template Description		
Rx timeout for the follow-up message.		
M2 Parameter		

SystemTemplate::GlobalTime::GlobalTimeSlave.followUpTimeoutValue	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00007]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeSlave
BSW Parameter	BSW Type
EthTSynRxCrcValidated	EcucEnumerationParamDef
BSW Description	
Definition of whether or not validation of the CRC takes place.	
Template Description	
Definition of whether or not validation of the CRC is supported.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::GlobalTimeEthSlave.crcValidated	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00049]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeSlave/EthTSynRxCrcValidated
BSW Parameter	BSW Type
CRC_IGNORED	EcucEnumerationLiteralDef
BSW Description	
EthTSyn ignores any CRC inside the Sub-TLVs.	
Template Description	
The CRC is supposed to be ignored	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcIgnored	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/EthTSynGlobalTimeSlave/EthTSynRxCrcValidated
BSW Parameter	BSW Type
CRC_NOT_VALIDATED	EcucEnumerationLiteralDef
BSW Description	
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60.	
Template Description	
The CRC is not supposed to be present. If CRC is present the message is ignored.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcNotValidated	

Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynRxCrcValidated	
BSW Parameter	BSW Type	
CRC_OPTIONAL	EcucEnumerationLiteralDef	
BSW Description		
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value.		
Template Description		
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcOptional		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave/EthTSynRxCrcValidated	
BSW Parameter	BSW Type	
CRC_VALIDATED	EcucEnumerationLiteralDef	
BSW Description		
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value. EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.		
Template Description		
This CRC is supposed to be validated.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcValidated		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig/EthTSynPortRole/ EthTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
EthTSynRxTmacValidated	EcucEnumerationParamDef	
BSW Description		
This parameter controls whether or not TMAC validation shall be supported.		
Template Description		

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00080]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPortConfig
BSW Parameter	BSW Type
EthTSynSwitchManagementEthSwitchPortRef	EcucReferenceDef
BSW Description	
In an AVB-Bridge config, this reference is used to assign the EthTSyn-Port to an Ethernet Switch-Port.	
Template Description	
Defines which CouplingPort is managed by this EthGlobalTimeManagedCouplingPort.	
M2 Parameter	
SystemTemplate::GlobalTime::ETH::EthGlobalTimeManagedCouplingPort.couplingPort	
Mapping Rule	Mapping Type
If the referenced CouplingPort is aggregated by a CouplingElement with couplingType = switch then the reference EthTSynSwitchManagementEthSwitchPortRef shall be defined and refer to the EthSwTPort which was derived from the CouplingPort.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00066]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	BSW Type
EthTSynSynchronizedTimeBaseRef	EcucReferenceDef
BSW Description	
Mandatory reference to the required synchronized time-base.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00013]

C.8.4 Flexray Time Management

BSW Module	BSW Context
FrTSyn	FrTSyn
BSW Parameter	BSW Type
FrTSynGeneral	EcucParamConfContainerDef
BSW Description	
This container holds the general parameters of the Flexray-specific Synchronized Time-base Manager	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00003]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGeneral
BSW Parameter	BSW Type
FrTSynDevErrorDetect	EcucBooleanParamDef
BSW Description	
Switches the development error detection and notification on or off. * true: detection and notification is enabled. * false: detection and notification is disabled.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00002]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGeneral
BSW Parameter	BSW Type
FrTSynMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Schedule period of the main function FrTSyn_MainFunction. Unit: [s].	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00016]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGeneral
BSW Parameter	BSW Type
FrTSynTimeValidationSupport	EcucBooleanParamDef
BSW Description	
Switches support for Time Validation on or off. - true: Time Validation is enabled. - false: Time Validation is disabled.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00040]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGeneral
BSW Parameter	BSW Type
FrTSynVersionInfoApi	EcucBooleanParamDef
BSW Description	
Activate/Deactivate the version information API (FrTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00019]

BSW Module	BSW Context
FrTSyn	FrTSyn
BSW Parameter	BSW Type
FrTSynGlobalTimeDomain	EcucParamConfContainerDef
BSW Description	
This represents the existence of a global time domain on Flexray. The FrTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains.	
If the FrTSyn exists it is assumed that at least one global time domain exists.	
Template Description	
This represents the ability to define a global time domain.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain	
Mapping Rule	Mapping Type
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the Ecuc Instance for which the ECU configuration is created.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00004]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	BSW Type
FrTSynEnableTimeValidation	EcucBooleanParamDef

BSW Description	
Enables/disables time recording for Time Validation for a specific Time Domain.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00041]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	BSW Type
FrTSynGlobalTimeDomainId	EcucIntegerParamDef
BSW Description	
The global time domain ID.	
Template Description	
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.domainId	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00005]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	BSW Type
FrTSynGlobalTimeMaster	EcucParamConfContainerDef
BSW Description	
Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.	
Template Description	
This represents the generic concept of a global time master.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster	
Mapping Rule	Mapping Type
The existence of the FrTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn is aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00006]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynCyclicMsgResumeTime	EcucFloatParamDef
BSW Description	

Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds	
Template Description	
Defines the minimum time between an "immediate" message and the next periodic message.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster.immediateResumeTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00032]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynGlobalTimeDebounceTime	EcucFloatParamDef
BSW Description	
This represents the configuration of a TX debounce time for SYNC and OFS messages compared to a message before with the same PDU. Unit: seconds	
Template Description	
Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.debounceTime	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00033]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynGlobalTimeMasterPdu	EcucParamConfContainerDef
BSW Description	
This container carries all properties required to configure the PDU sent by the global time master for the given global time domain.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00008]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeMasterPdu
BSW Parameter	BSW Type
FrTSynGlobalTimeMasterHandleId	EcucIntegerParamDef
BSW Description	
This represents the handle ID of the PDU that contains the global time information.	

Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00007]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeMasterPdu
BSW Parameter	BSW Type
FrTSynGlobalTimePduRef	EcucReferenceDef
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.pduTriggering	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynGlobalTimeTx_crcSecured	EcucEnumerationParamDef
BSW Description	
This represents the configuration of whether or not CRC is supported.	
Template Description	
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.	
M2 Parameter	
SystemTemplate::GlobalTime::FR::GlobalTimeFrMaster.crcSecured	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00013]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTx_crcSecured
BSW Parameter	BSW Type
CRC_NOT_SUPPORTED	EcucEnumerationLiteralDef
BSW Description	
This represents a configuration where CRC is not supported.	

Template Description	
This indicates that CRC is not supported	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcNotSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTxCrcSecured
BSW Parameter	BSW Type
CRC_SUPPORTED	EcucEnumerationLiteralDef
BSW Description	
This represents a configuration where CRC is supported.	
Template Description	
This indicates that CRC is supported	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcSupportEnum.crcSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynGlobalTimeTxPeriod	EcucFloatParamDef
BSW Description	
This represents the TX period. Unit: seconds	
Template Description	
This represents the period. Unit: seconds	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeMaster.syncPeriod	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00014]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	BSW Type
FrTSynImmediateTimeSync	EcucBooleanParamDef
BSW Description	
Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within FrTSyn_MainFunction().	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00031]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster	
BSW Parameter	BSW Type	
FrTSynTxTmacCalculated	EcucEnumerationParamDef	
BSW Description		
This parameter controls whether or not TMAC calculation shall be supported.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00036]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain	
BSW Parameter	BSW Type	
FrTSynGlobalTimeOfsDataIDList	EcucParamConfContainerDef	
BSW Description		
The DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
The DataIDList for OFS messages to calculate CRC.		
M2 Parameter		
SystemTemplate::GlobalTime::FR::FrGlobalTimeDomainProps.ofsDataIDList		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00024]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeOfsDataIDList	
BSW Parameter	BSW Type	
FrTSynGlobalTimeOfsDataIDListElement	EcucParamConfContainerDef	
BSW Description		
Element of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
The DataIDList for OFS messages to calculate CRC.		
M2 Parameter		
SystemTemplate::GlobalTime::FR::FrGlobalTimeDomainProps.ofsDataIDList		
Mapping Rule		Mapping Type
Value shall be derived from element of the ordered ofsDataIDList.		full

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00028]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeOfsDataIDList/FrTSynGlobalTimeOfsDataIDListElement	
BSW Parameter		BSW Type
FrTSynGlobalTimeOfsDataIDListIndex		EcucIntegerParamDef
BSW Description		
Index of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTSyn_00029]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeOfsDataIDList/FrTSynGlobalTimeOfsDataIDListElement	
BSW Parameter		BSW Type
FrTSynGlobalTimeOfsDataIDListValue		EcucIntegerParamDef
BSW Description		
Value of the DataIDList for OFS messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local
Mapping Status		ECUC Parameter ID
valid		[ECUC_FrTSyn_00030]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain	
BSW Parameter		BSW Type
FrTSynGlobalTimeSecureTmacLength		EcucIntegerParamDef
BSW Description		
Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used.		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
		local

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00034]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain	
BSW Parameter	BSW Type	
FrTSynGlobalTimeSlave	EcucParamConfContainerDef	
BSW Description		
This represents the time slave for the enclosing global time domain.		
Template Description		
This represents the generic concept of a global time slave.		
M2 Parameter		
SystemTemplate::GlobalTime::GlobalTimeSlave		
Mapping Rule	Mapping Type	
The existence of the FrTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn is aggregated at the EcuInstance for which the ECU configuration exists,	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00010]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
FrTSynGlobalTimeMinMsgGap	EcucFloatParamDef	
BSW Description		
This parameter represents the configuration of a minimum message gap time for received SYNC and OFS messages compared to a message before with the same PDU. If PDUs are received more often in between than this parameter allows, they shall be ignored.		
Unit: seconds		
Template Description		
M2 Parameter		
Mapping Rule	Mapping Type	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00038]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave	
BSW Parameter	BSW Type	
FrTSynGlobalTimeSequenceCounterJumpWidth	EcucIntegerParamDef	
BSW Description		
The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC resp. two OFS messages.		
Template Description		
Specifies the maximum allowed gap of the sequence counter between two SYNC resp. two OFS messages.		
M2 Parameter		

SystemTemplate::GlobalTime::FR::GlobalTimeFrSlave.sequenceCounterJumpWidth	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00022]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	BSW Type
FrTSynGlobalTimeSlavePdu	EcucParamConfContainerDef
BSW Description	
This container carries all properties required to configure the PDU received by the time slave for the given global time domain.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00012]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeSlavePdu
BSW Parameter	BSW Type
FrTSynGlobalTimePduRef	EcucReferenceDef
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeDomain.pduTriggering	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeSlavePdu
BSW Parameter	BSW Type
FrTSynGlobalTimeSlaveHandleId	EcucIntegerParamDef
BSW Description	
This represents the handle ID of the PDU that contains the global time information.	
Template Description	

M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00011]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	BSW Type
FrTSynGlobalTimeTmacTimeout	EcucFloatParamDef
BSW Description	
Rx timeout for the TMAC message.	
Unit:seconds	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00039]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	BSW Type
FrTSynRxCrcValidated	EcucEnumerationParamDef
BSW Description	
This parameter controls whether or not CRC validation shall be supported.	
Template Description	
Definition of whether or not validation of the CRC is supported.	
M2 Parameter	
SystemTemplate::GlobalTime::FR::GlobalTimeFrSlave.crcValidated	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00017]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrcValidated
BSW Parameter	BSW Type
CRC_IGNORED	EcucEnumerationLiteralDef
BSW Description	
The Timesync module accepts Time Synchronization messages, which are CRC secured (without actually validating the CRC) and those, which are not CRC secured. That means, the Timesync module ignores the CRC.	
Template Description	
The CRC is supposed to be ignored	

M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcIgnored	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrc Validated
BSW Parameter	BSW Type
CRC_NOT_VALIDATED	EcucEnumerationLiteralDef
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are not CRC secured. All other Time Synchronization messages are ignored.	
Template Description	
The CRC is not supposed to be present. If CRC is present the message is ignored.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcNotValidated	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrc Validated
BSW Parameter	BSW Type
CRC_OPTIONAL	EcucEnumerationLiteralDef
BSW Description	
The Timesync module accepts only Time Synchronization messages which are not CRC secured and Time Synchronization messages which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Template Description	
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcOptional	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrc Validated
BSW Parameter	BSW Type
CRC_VALIDATED	EcucEnumerationLiteralDef
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	

Template Description	
This CRC is supposed to be validated.	
M2 Parameter	
SystemTemplate::GlobalTime::GlobalTimeCrcValidationEnum.crcValidated	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	BSW Type
FrTSynRxTmacValidated	EcucEnumerationParamDef
BSW Description	
This parameter controls whether or not TMAC validation shall be supported.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00037]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	BSW Type
FrTSynGlobalTimeSyncDataIDList	EcucParamConfContainerDef
BSW Description	
The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for SYNC messages to calculate CRC.	
M2 Parameter	
SystemTemplate::GlobalTime::FR::FrGlobalTimeDomainProps.syncDataIDList	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00023]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSyncDataIDList
BSW Parameter	BSW Type
FrTSynGlobalTimeSyncDataIDListElement	EcucParamConfContainerDef
BSW Description	
Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for SYNC messages to calculate CRC.	
M2 Parameter	

SystemTemplate::GlobalTime::FR::FrGlobalTimeDomainProps.syncDataIDList	
Mapping Rule	Mapping Type
Value shall be derived from element of the ordered syncDataIDList.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00025]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSyncDataIDList/FrTSynGlobalTimeSyncDataIDListElement
BSW Parameter	BSW Type
FrTSynGlobalTimeSyncDataIDListIndex	EcucIntegerParamDef
BSW Description	
Index of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00026]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSyncDataIDList/FrTSynGlobalTimeSyncDataIDListElement
BSW Parameter	BSW Type
FrTSynGlobalTimeSyncDataIDListValue	EcucIntegerParamDef
BSW Description	
Value of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
M2 Parameter	
Mapping Rule	Mapping Type
	local
Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00027]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	BSW Type
FrTSynSynchronizedTimeBaseRef	EcucReferenceDef
BSW Description	
Mandatory reference to the required synchronized time-base.	
Template Description	
M2 Parameter	

Mapping Rule		Mapping Type
		local
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00018]	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain	
BSW Parameter		BSW Type
FrTSynUseExtendedMsgFormat		EcucBooleanParamDef
BSW Description		
* true: use at least 32 byte for Timesync messages (depending on configuration)		
* false: use always 16 byte for Timesync messages		
Template Description		
M2 Parameter		
Mapping Rule		Mapping Type
Mapping Status	ECUC Parameter ID	
valid	[ECUC_FrTSyn_00035]	

C.9 Crypto Stack

C.9.1 CryptoDriver

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter		BSW Type
CRYPTO_ALGOFAM_3DES		EcucEnumerationLiteralDef
BSW Description		
0x13		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule		Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_3DES		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter		BSW Type
CRYPTO_ALGOFAM_AES		EcucEnumerationLiteralDef
BSW Description		
0x14		
Template Description		

This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_AES	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_1_256	EcucEnumerationLiteralDef
BSW Description	
0x0F	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_1_512	EcucEnumerationLiteralDef
BSW Description	
0x10	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_2s_256	EcucEnumerationLiteralDef
BSW Description	
0x11	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	

M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1s_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_2s_512	EcucEnumerationLiteralDef
BSW Description	
0x12	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_2s_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BRAINPOOL	EcucEnumerationLiteralDef
BSW Description	
0x18	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BRAINPOOL	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_CHACHA	EcucEnumerationLiteralDef
BSW Description	
0x15	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	

Mapping Rule		Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_CHACHA		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_CUSTOM	EcucEnumerationLiteralDef	
BSW Description	0xFF	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Custom algorithm specified	Mapping Type
		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_ECCNIST	EcucEnumerationLiteralDef	
BSW Description	0x19	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	if algorithmFamily == CRYPTO_ALGOFAM_ECCNIST	Mapping Type
		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_ED25519	EcucEnumerationLiteralDef	
BSW Description	0x17	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	if algorithmFamily == CRYPTO_ALGOFAM_ED25519	Mapping Type
		full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_NOT_SET	EcucEnumerationLiteralDef	
BSW Description		
0x00		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
If algorithmFamily is not defined.	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_RIPEMD160	EcucEnumerationLiteralDef	
BSW Description		
0x0E		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_RIPEMD160	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_RNG	EcucEnumerationLiteralDef	
BSW Description		
0x1B		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_RNG	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_RSA	EcucEnumerationLiteralDef	
BSW Description	0x16	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_RSA	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_SHA1	EcucEnumerationLiteralDef	
BSW Description	0x01	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type	
If algorithmFamily == CRYPTO_ALGOFAM_SHA1	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_SHA2_224	EcucEnumerationLiteralDef	
BSW Description	0x02	
Template Description	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type	
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_224	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily	

BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_256	EcucEnumerationLiteralDef
BSW Description	
0x03	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_384	EcucEnumerationLiteralDef
BSW Description	
0x04	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_384	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_512	EcucEnumerationLiteralDef
BSW Description	
0x05	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_512_224	EcucEnumerationLiteralDef

BSW Description	
0x06	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_512_224	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_512_256	EcucEnumerationLiteralDef
BSW Description	
0x07	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA2_512_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_224	EcucEnumerationLiteralDef
BSW Description	
0x08	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_224	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_256	EcucEnumerationLiteralDef
BSW Description	
0x09	

Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_384	EcucEnumerationLiteralDef
BSW Description	
0x0A	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_384	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_512	EcucEnumerationLiteralDef
BSW Description	
0x0B	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHAKE128	EcucEnumerationLiteralDef
BSW Description	
0x0C	
Template Description	

This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHAKE128	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHAKE256	EcucEnumerationLiteralDef
BSW Description	
0x0D	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHAKE256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmFamily
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SIPHASH	EcucEnumerationLiteralDef
BSW Description	
0x1C	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SIPHASH	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_12ROUNDS	EcucEnumerationLiteralDef
BSW Description	
0x0D	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	

M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_12ROUNDS	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_20ROUNDS	EcucEnumerationLiteralDef
BSW Description	
0x0E	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_20ROUNDS	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_8ROUNDS	EcucEnumerationLiteralDef
BSW Description	
0x0C	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_8ROUNDS	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_CBC	EcucEnumerationLiteralDef
BSW Description	
0x02	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	

Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_CBC		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_CFB		EcucEnumerationLiteralDef
BSW Description		
0x03		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_CFB		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_CMAC		EcucEnumerationLiteralDef
BSW Description		
0x10		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_CMAC		full
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_CTR		EcucEnumerationLiteralDef
BSW Description		
0x05		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_CTR		full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_CTRDRBG	EcucEnumerationLiteralDef	
BSW Description		
0x12		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule	Mapping Type	
if algorithmMode == CRYPTO_ALGOMODE_CTRDRBG	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_ECB	EcucEnumerationLiteralDef	
BSW Description		
0x01		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule	Mapping Type	
if algorithmMode == CRYPTO_ALGOMODE_ECB	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_GCM	EcucEnumerationLiteralDef	
BSW Description		
0x06		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule	Mapping Type	
if algorithmMode == CRYPTO_ALGOMODE_GCM	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_GMAC	EcucEnumerationLiteralDef	
BSW Description	0x11	
Template Description	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type	
if algorithmMode == CRYPTO_ALGOMODE_GMAC	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_HMAC	EcucEnumerationLiteralDef	
BSW Description	0x0F	
Template Description	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type	
if algorithmMode == CRYPTO_ALGOMODE_HMAC	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter	BSW Type	
CRYPTO_ALGOMODE_NOT_SET	EcucEnumerationLiteralDef	
BSW Description	0x00	
Template Description	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type	
if algorithmMode is not set	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	

BSW Parameter		BSW Type
CRYPTO_ALGOMODE_OFB		EcucEnumerationLiteralDef
BSW Description		
0x04		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_OFB		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_PXXXR		EcucEnumerationLiteralDef
BSW Description		
0x09		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_RSAES_OAEP		EcucEnumerationLiteralDef
BSW Description		
0x08		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_RSAES_OAEP		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode	
BSW Parameter		BSW Type
CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5		EcucEnumerationLiteralDef

BSW Description	
0x09	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5	EcucEnumerationLiteralDef
BSW Description	
0x0B	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_RSASSA_PSS	EcucEnumerationLiteralDef
BSW Description	
0x0A	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_RSASSA_PSS	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_SIPHASH_2_4	EcucEnumerationLiteralDef
BSW Description	
0x13	

Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_SIPHASH_2_4	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_SIPHASH_4_8	EcucEnumerationLiteralDef
BSW Description	
0x14	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_SIPHASH_4_8	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmMode
BSW Parameter	BSW Type
CRYPTO_ALGOMODE_XTS	EcucEnumerationLiteralDef
BSW Description	
0x07	
Template Description	
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type
if algorithmMode == CRYPTO_ALGOMODE_XTS	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_3DES	EcucEnumerationLiteralDef
BSW Description	
0x13	
Template Description	

This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_3DES	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_AES	EcucEnumerationLiteralDef
BSW Description	
0x14	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_AES	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_1_256	EcucEnumerationLiteralDef
BSW Description	
0x0F	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_1_512	EcucEnumerationLiteralDef
BSW Description	
0x10	

Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_2s_256	EcucEnumerationLiteralDef
BSW Description	
0x11	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_1s_256	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BLAKE_2s_512	EcucEnumerationLiteralDef
BSW Description	
0x12	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BLAKE_2s_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_BRAINPOOL	EcucEnumerationLiteralDef
BSW Description	

0x18	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_BRAINPOOL	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_CHACHA	EcucEnumerationLiteralDef
BSW Description	
0x15	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_CHACHA	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_CUSTOM	EcucEnumerationLiteralDef
BSW Description	
0xFF	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
Custom algorithm specified	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_ECCNIST	EcucEnumerationLiteralDef

BSW Description	
0x19	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_ECCNIST	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_ED25519	EcucEnumerationLiteralDef	
BSW Description		
0x17		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_ED25519	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_NOT_SET	EcucEnumerationLiteralDef	
BSW Description		
0x00		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
If algorithmFamily is not defined.	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter	BSW Type	

CRYPTO_ALGOFAM_RIPEMD160	EcucEnumerationLiteralDef
BSW Description	
0x0E	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_RIPEMD160	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_RNG	EcucEnumerationLiteralDef	
BSW Description		
0x1B		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_RNG	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter	BSW Type	
CRYPTO_ALGOFAM_RSA	EcucEnumerationLiteralDef	
BSW Description		
0x16		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule	Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_RSA	full	
Mapping Status	ECUC Parameter ID	
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	

BSW Parameter		BSW Type
CRYPTO_ALGOFAM_SHA1		EcucEnumerationLiteralDef
BSW Description		
0x01		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule		Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA1		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter		BSW Type
CRYPTO_ALGOFAM_SHA2_224		EcucEnumerationLiteralDef
BSW Description		
0x02		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule		Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_224		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	
BSW Parameter		BSW Type
CRYPTO_ALGOFAM_SHA2_256		EcucEnumerationLiteralDef
BSW Description		
0x03		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule		Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_256		full
Mapping Status		ECUC Parameter ID
valid		

BSW Module	BSW Context	
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family	

BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_384	EcucEnumerationLiteralDef
BSW Description	
0x04	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_384	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_512	EcucEnumerationLiteralDef
BSW Description	
0x05	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA2_512_224	EcucEnumerationLiteralDef
BSW Description	
0x06	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
If algorithmFamily == CRYPTO_ALGOFAM_SHA2_512_224	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family

BSW Parameter		BSW Type	
CRYPTO_ALGOFAM_SHA2_512_256		EcucEnumerationLiteralDef	
BSW Description			
0x07			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_SHA2_512_256		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module	BSW Context		
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family		
BSW Parameter		BSW Type	
CRYPTO_ALGOFAM_SHA3_224		EcucEnumerationLiteralDef	
BSW Description			
0x08			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_224		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module	BSW Context		
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family		
BSW Parameter		BSW Type	
CRYPTO_ALGOFAM_SHA3_256		EcucEnumerationLiteralDef	
BSW Description			
0x09			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_256		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module	BSW Context		
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family		

BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_384	EcucEnumerationLiteralDef
BSW Description	
0x0A	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_384	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHA3_512	EcucEnumerationLiteralDef
BSW Description	
0x0B	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHA3_512	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family
BSW Parameter	BSW Type
CRYPTO_ALGOFAM_SHAKE128	EcucEnumerationLiteralDef
BSW Description	
0x0C	
Template Description	
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.	
M2 Parameter	
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily	
Mapping Rule	Mapping Type
if algorithmFamily == CRYPTO_ALGOFAM_SHAKE128	full
Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family

BSW Parameter		BSW Type	
CRYPTO_ALGOFAM_SHAKE256		EcucEnumerationLiteralDef	
BSW Description			
0x0D			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_SHAKE256		full	
Mapping Status		ECUC Parameter ID	
valid			

BSW Module	BSW Context		
Crypto	Crypto/CryptoPrimitives/CryptoPrimitive/CryptoPrimitiveAlgorithmSecondary Family		
BSW Parameter		BSW Type	
CRYPTO_ALGOFAM_SIPHASH		EcucEnumerationLiteralDef	
BSW Description			
0x1C			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
if algorithmFamily == CRYPTO_ALGOFAM_SIPHASH		full	
Mapping Status		ECUC Parameter ID	
valid			

C.9.2 Crypto Service Manager

BSW Module	BSW Context		
Csm	Csm/CsmJobs		
BSW Parameter		BSW Type	
CsmJob		EcucParamConfContainerDef	
BSW Description			
Container for configuration of CSM job. The container name serves as a symbolic name for the identifier of a job configuration.			
Template Description			
This meta-class has the ability to represent a crypto service mapping for the Pdu-based communication via SecOC.			
M2 Parameter			
SystemTemplate::SecureCommunication::SecOcCryptoServiceMapping			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00118]	

BSW Module		BSW Context	
Csm		Csm/CsmJobs/CsmJob	
BSW Parameter		BSW Type	
CsmJobKeyRef		EcucReferenceDef	
BSW Description			
This parameter refers to the key which shall be used for the CsmPrimitive. It's possible to use a CsmKey for different jobs			
Template Description			
This meta-class has the ability to represent a crypto key			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServiceKey			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00126]	

BSW Module		BSW Context	
Csm		Csm/CsmJobs/CsmJob	
BSW Parameter		BSW Type	
CsmJobPrimitiveRef		EcucReferenceDef	
BSW Description			
This parameter refers to the used CsmPrimitive. Different jobs may refer to one CsmPrimitive. The referred CsmPrimitive provides detailed information on the actual cryptographic routine.			
Template Description			
This reference identifies the applicable crypto primitive for the authentication.			
M2 Parameter			
SystemTemplate::SecureCommunication::SecOcCryptoServiceMapping.authentication			
Mapping Rule			Mapping Type
1:1 mapping			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00122]	

BSW Module		BSW Context	
Csm		Csm/CsmJobs/CsmJob	
BSW Parameter		BSW Type	
CsmJobQueueRef		EcucReferenceDef	
BSW Description			
This parameter refers to the queue. The queue is used if the underlying crypto driver object is busy. The queue refers also to the channel which is used.			
Template Description			
This meta-class has the ability to represent a crypto queue.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServiceQueue			
Mapping Rule			Mapping Type
If the SecOcCryptoServiceMapping this CsmJob is derived from has a reference cryptoServiceQueue defined then the CsmJobQueueRef shall refer to the respective CsmQueue.			full
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00125]	

BSW Module		BSW Context	
Csm		Csm/CsmKeys	
BSW Parameter		BSW Type	
CsmKey		EcucParamConfContainerDef	
BSW Description			
Container for configuration of a CSM key. The container name serves as a symbolic name for the identifier of a key configuration.			
Template Description			
This meta-class has the ability to represent a crypto key			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServiceKey			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00014]	

BSW Module		BSW Context	
Csm		Csm/CsmPrimitives/CsmMacGenerate/CsmMacGenerateConfig	
BSW Parameter		BSW Type	
CsmMacGenerateAlgorithmFamily		EcucEnumerationParamDef	
BSW Description			
Determines the algorithm family used for the crypto service. This parameter defines the most significant part of the algorithm.			
Template Description			
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00188]	

BSW Module		BSW Context	
Csm		Csm/CsmPrimitives/CsmMacGenerate/CsmMacGenerateConfig	
BSW Parameter		BSW Type	
CsmMacGenerateAlgorithmKeyLength		EcucIntegerParamDef	
BSW Description			
Size of the MAC key in bytes			
Template Description			
This attribute describes the length of the cryptographic key.			
M2 Parameter			
SystemTemplate::SecureCommunication::CryptoServiceKey.length			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		ECUC Parameter ID	
valid		[ECUC_Csm_00044]	

BSW Module		BSW Context	
Csm		Csm/CsmPrimitives/CsmMacGenerate/CsmMacGenerateConfig	

BSW Parameter		BSW Type
CsmMacGenerateAlgorithmMode		EcucEnumerationParamDef
BSW Description		
Determines the algorithm mode used for the crypto service		
Template Description		
This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Csm_00189]

BSW Module	BSW Context	
Csm	Csm/CsmPrimitives/CsmMacGenerate/CsmMacGenerateConfig	
BSW Parameter		BSW Type
CsmMacGenerateAlgorithmSecondaryFamily		EcucEnumerationParamDef
BSW Description		
Determines the secondary algorithm family used for the crypto service		
Template Description		
This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive.		
The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmSecondaryFamily		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Csm_00134]

BSW Module	BSW Context	
Csm	Csm/CsmPrimitives/CsmMacVerify/CsmMacVerifyConfig	
BSW Parameter		BSW Type
CsmMacVerifyAlgorithmFamily		EcucEnumerationParamDef
BSW Description		
Determines the algorithm family used for the crypto service. This parameter defines the most significant part of the algorithm.		
Template Description		
This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmFamily		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Csm_00051]

BSW Module	BSW Context	
Csm	Csm/CsmPrimitives/CsmMacVerify/CsmMacVerifyConfig	
BSW Parameter	BSW Type	
CsmMacVerifyAlgorithmKeyLength	EcucIntegerParamDef	
BSW Description	Size of the MAC key in bytes	
Template Description	This attribute describes the length of the cryptographic key.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServiceKey.length	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Csm_00193]	

BSW Module	BSW Context	
Csm	Csm/CsmPrimitives/CsmMacVerify/CsmMacVerifyConfig	
BSW Parameter	BSW Type	
CsmMacVerifyAlgorithmMode	EcucEnumerationParamDef	
BSW Description	Determines the algorithm mode used for the crypto service	
Template Description	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmMode	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Csm_00195]	

BSW Module	BSW Context	
Csm	Csm/CsmPrimitives/CsmMacVerify/CsmMacVerifyConfig	
BSW Parameter	BSW Type	
CsmMacVerifyAlgorithmSecondaryFamilyCustom	EcucStringParamDef	
BSW Description	This is the second the name of the custom algorithm, if CRYPTO_ALGOFAM_CUSTOM is set as CsmMacVerifyAlgorithmSecondaryFamily	
Template Description	This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive. The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.	
M2 Parameter	SystemTemplate::SecureCommunication::CryptoServicePrimitive.algorithmSecondaryFamily	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	ECUC Parameter ID	
valid	[ECUC_Csm_00141]	

BSW Module	BSW Context	
Csm	Csm/CsmQueues	
BSW Parameter		BSW Type
CsmQueue		EcucParamConfContainerDef
BSW Description		
Container for configuration of a CSM queue. A queue has two tasks: <ol style="list-style-type: none"> 1. queue jobs which cannot be processed since the underlying hardware is busy and 2. refer to channel which shall be used 		
Template Description		
This meta-class has the ability to represent a crypto queue.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServiceQueue		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Csm_00032]

BSW Module	BSW Context	
Csm	Csm/CsmQueues/CsmQueue	
BSW Parameter		BSW Type
CsmQueueSize		EcucIntegerParamDef
BSW Description		
Size of the CsmQueue. If jobs cannot be processed by the underlying hardware since the hardware is busy, the jobs stay in the prioritized queue. If the queue is full, the next job will be rejected.		
Template Description		
Defines the queue size of the CryptoServiceQueue.		
M2 Parameter		
SystemTemplate::SecureCommunication::CryptoServiceQueue.queueSize		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Csm_00034]

C.10 Services

C.10.1 Transformer General

BSW Module	BSW Context	
Xfrm	Xfrm/XfrmImplementationMapping/XfrmSignal/XfrmSignalChoice/XfrmISignalGroupRefChoice	
BSW Parameter		BSW Type
XfrmISignalGroupRef		EcucForeignReferenceDef
BSW Description		
Reference to the ISignalGroup in the system description that transports the transformed data.		
Template Description		

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 ISignal-Group represents a COM Signal Group.	
Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalGroup	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xfrm_00010]

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping/XfrmSignal/XfrmSignalChoice/XfrmISignalRefChoice
BSW Parameter	BSW Type
XfrmISignalRef	EcucForeignReferenceDef
BSW Description	
Reference to the ISignal in the system description that transports the transformed data.	
Template Description	
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.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xfrm_00008]

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping
BSW Parameter	BSW Type
XfrmTransformationTechnologyRef	EcucForeignReferenceDef
BSW Description	
Reference to the TransformationTechnology in the DataTransformation of the system description for which the implementation (BswModuleEntry) shall be mapped.	
Template Description	
A TransformationTechnology is a transformer inside a transformer chain.	
M2 Parameter	

SystemTemplate::Transformer::TransformationTechnology	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xfrm_00003]

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping
BSW Parameter	BSW Type
XfrmVariableDataPrototypeInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to a VariableDataPrototype in case a dedicated transformer BswModuleEntry is required per VariableDataPrototype access.	
Template Description	
This attribute defines whether the Transformer has an internal state or not.	
M2 Parameter	
SystemTemplate::Transformer::TransformationTechnology.hasInternalState	
Mapping Rule	Mapping Type
If an ISignal with dataTypePolicy set to transformingISignal - is received by this EcucInstance and - one of the TransformerTechnologies in the transformerChain has TransformationTechnology.hasInternalState set to true and - the corresponding VariableDataPrototype is consumed via multiple RPorts then this XfrmVariableDataPrototypeInstanceRef shall be used.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Xfrm_00011]

C.11 Miscellaneous

C.11.1 Log and Trace

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel
BSW Parameter	BSW Type
DltTxPdu	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters of the AUTOSAR Dlt module's Tx Pdus.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::GeneralPurposeIPdu	
Mapping Rule	Mapping Type
This container shall be created if a GeneralPurposeIPdu with the category DLT is defined in the ECU_SYSTEM_DESCRIPTION and this GeneralPurposeIPdu is transmitted by the regarded Ecu.	full
Mapping Status	ECUC Parameter ID
valid	[ECUC_Dlt_00907]

BSW Module	BSW Context
------------	-------------

Dlt	Dlt/DltConfigSet	
BSW Parameter		BSW Type
DltRxPdu	EcucParamConfContainerDef	
BSW Description		
Contains the Pdu IDs to be used for Dlt control messages reception.		
Template Description		
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::GeneralPurposeIPdu		
Mapping Rule		Mapping Type
This container shall be created if a GeneralPurposeIPdu with the category DLT is defined in the ECU_SYSTEM_DESCRIPTION and this GeneralPurposeIPdu is received by the regarded Ecu.		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_Dlt_00900]

C.11.2 ECU State Manager

BSW Module	BSW Context	
EcuM	EcuM/EcuMConfiguration/EcuMCommonConfiguration	
BSW Parameter		BSW Type
EcuMWakeupSource	EcucParamConfContainerDef	
BSW Description		
These containers describe the configured wakeup sources.		
Template Description		
If this parameter is available and set to true then a channel wakeup source shall be created for the PhysicalChannel referencing this CommunicationConnector.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.createEcuWakeupSource		
Mapping Rule		Mapping Type
1:1 Mapping to SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.createEcuWakeupSource == True		full
Mapping Status		ECUC Parameter ID
valid		[ECUC_EcuM_00150]

BSW Module	BSW Context	
EcuM	EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource	
BSW Parameter		BSW Type
EcuMComMPNCTRef	EcucReferenceDef	
BSW Description		
This is a reference to a one or more PNC's defined in the Communication Manager.		
No reference indicates that the wakeup source is not assigned to a partial network.		
Template Description		
If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs.		
This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.pncSynchronousWakeup		

Mapping Rule		Mapping Type
if SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.pncSynchronousWakeup == True: Add reference for each SystemTemplate::PncMapping::PncMapping otherwise: Add reference for each SystemTemplate::PncMapping::PncMapping where pncWakeupEnable == True and which is connected to the according PhysicalChannel		full
Mapping Status	ECUC Parameter ID	
valid	[ECUC_EcuM_00228]	

D Constraint History

D.1 Constraint History of this Document according to AUTOSAR R4.0.1

D.1.1 Changed Constraints in R4.0.1

N/A

D.1.2 Added Constraints in R4.0.1

Number	Heading
[constr_3000]	valid SenderRecCompositeTypeMappings
[constr_3001]	valid ClientServerToSignalGroupMappings
[constr_3002]	valid SwcToImplMapping
[constr_3003]	Number of CAN channels
[constr_3004]	Clustering and separation must be exclusive
[constr_3005]	valid EcuResourceEstimation
[constr_3006]	valid EcuMapping
[constr_3007]	SelectorFieldCodes for dynamic part alternatives
[constr_3008]	EcuInstance subelements
[constr_3009]	Overlapping of ISignals is prohibited
[constr_3010]	ISignalIPdu shall not be exceeded
[constr_3011]	Overlapping of updateIndicationBits for ISignals is prohibited
[constr_3012]	Overlapping of Pdus is prohibited
[constr_3013]	Frame length shall not be exceeded
[constr_3014]	Overlapping of updateIndicationBits for Pdus is prohibited
[constr_3015]	Number of LIN channels
[constr_3016]	Number of Ethernet channels
[constr_3017]	Length of multiplexed Pdu shall not be exceeded
[constr_3018]	Number of FlexRay channels

Table D.1: Added Constraints in R4.0.1

D.1.3 Deleted Constraints in R4.0.1

N/A

D.2 Constraint History of this Document according to AUTOSAR R4.0.2

D.2.1 Changed Constraints in R4.0.2

N/A

D.2.2 Added Constraints in R4.0.2

Number	Heading
[constr_3019]	In the flat ECU extract each required interface must be satisfied by connected provided interfaces

Table D.2: Added Constraints in R4.0.2

D.2.3 Deleted Constraints in R4.0.2

N/A

D.3 Constraint and Specification Item History of this document according to AUTOSAR R4.0.3

D.3.1 Changed Constraints in R4.0.3

N/A

D.3.2 Changed Specification Items in R4.0.3

N/A

D.3.3 Added Constraints in R4.0.3

Number	Heading
[constr_3020]	CommunicationDirection of containedIPduGroups
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3024]	Usage of triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition is not allowed for signal groups and group signals.
[constr_3025]	Usage of NPdus in TpConnections
[constr_3026]	valid EmptySignalMappings

Table D.3: Added Constraints in R4.0.3

D.3.4 Added Specification Items in R4.0.3

Number	Heading
[TPS_SYST_01000]	FlatInstanceDescriptor roles

Table D.4: Added Specification Items in 4.0.3

D.3.5 Deleted Constraints in R4.0.3

N/A

D.3.6 Deleted Specification Items in R4.0.3

N/A

D.4 Constraint and Specification Item History of this document according to AUTOSAR R4.1.1

D.4.1 Changed Constraints in R4.1.1

Number	Heading
[constr_3018]	Number of FlexRay channels

Table D.5: Changed Constraints in R4.1.1

D.4.2 Changed Specification Items in R4.1.1

N/A

D.4.3 Added Constraints in R4.1.1

Number	Heading
[constr_1198]	TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping
[constr_1199]	ISignals relating to systemSignals eligible for a TriggerToSignalMapping
[constr_1206]	DataMapping to PRPortPrototype
[constr_1207]	Existence of the attribute DataMapping.communicationDirection in the context of a SenderReceiverInterface or TriggerInterface
[constr_1208]	Existence of the attribute DataMapping.communicationDirection in the context of a ClientServerInterface
[constr_1265]	DoIpGidSynchronizationNeeds can only exist once per ECU_EXTRACT
[constr_1266]	DoIpGidNeeds can only exist once per ECU_EXTRACT
[constr_1267]	DoIpActivationLineNeeds can only exist once per ECU_EXTRACT
[constr_3027]	Existence of ecuExtractVersion
[constr_3028]	FibexElements
[constr_3029]	Assign-Frame command usage
[constr_3030]	valid relationship between ECUMapping and EcuInstance
[constr_3031]	Complete System Description does not have outside ports
[constr_3032]	Combinations of SwcToEcuMapping targets
[constr_3033]	Criteria for primitive argument mapping

[constr_3034]	Values of LinSlaveConfig and LinSlave attributes
[constr_3035]	CanNm user data configuration in case NID/CBV are enabled
[constr_3036]	Pdus in CAN and LIN Frames
[constr_3037]	maximum Frame frameLength for CAN and LIN
[constr_3038]	maximum Frame frameLength for FlexRay
[constr_3039]	pnIdentifier range
[constr_3040]	Restriction of pnIdentifier values
[constr_3041]	pncVectorOffset range
[constr_3042]	pncVectorLength range
[constr_3043]	pncVector configuration in AUTOSAR Com
[constr_3044]	CBV configuration in case partial network is used
[constr_3045]	Signal content evaluation vs. Mode evaluation
[constr_3046]	Consistency of TransmissionModeCondition.iSignalInIPdu
[constr_3047]	Uniqueness of macMulticastAddresses
[constr_3048]	Range of vlanIdentifier
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs
[constr_3050]	J1939Cluster uses exactly one CanPhysicalChannel
[constr_3051]	Restriction of ISignalMapping references
[constr_3052]	Complete ISignalMapping of ISignalGroup signals
[constr_3053]	Complete ISignalMapping of target ISignalGroup
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3055]	SystemSignalGroup in a complete System Description
[constr_3056]	pduLength of the NmPdu
[constr_3057]	Maximal one BusspecificNmEcu per NmEcu and bus system is allowed to be defined
[constr_3058]	References from SenderRecArrayElementMapping and from Sender-RecRecordElementMapping to SystemSignals are not allowed within a SenderReceiverCompositeElementToSignalMapping
[constr_3059]	Mandatory DataMapping on the receiver side for elements of a composite data type
[constr_3060]	Usage of networkRepresentationProps and physicalProps
[constr_3061]	CompuMethod specification in networkRepresentationProps
[constr_3062]	The EcuInstance that is referenced from a specific CouplingElement shall be connected to the same EthernetCluster as the specific CouplingElement
[constr_3063]	Usage of portNumber and dynamicallyAssigned with value “true” is mutually exclusive
[constr_3064]	Usage of DataMapping.serviceInstance , DataMapping.eventHandler and DataMapping.eventGroup references
[constr_3065]	Mapping of queued Triggers to SystemSignals is prohibited
[constr_3066]	Restriction of SenderComSpecs that refer to dataElements mapped to the same SystemSignal
[constr_3067]	initValue defined in the context of ISignal
[constr_3068]	DoIPPowerModeStatusNeeds in the category ECU_EXTRACT
[constr_3501]	Role of SystemSignal in 1:n communication
[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3505]	Criteria for primitive Data Mapping
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3508]	Value of nmReadySleepTime
[constr_3514]	No two ISignalToIPduMappings shall reference the identical ISignal

Table D.6: Added Constraints in R4.1.1

D.4.4 Added Specification Items in R4.1.1

Number	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01002]	System Category
[TPS_SYST_01003]	Standardized System Category Definitions
[TPS_SYST_01004]	Definition of AUTOSAR ECU
[TPS_SYST_01005]	Definition of EcuInstance
[TPS_SYST_01006]	Assign ECU type to EcuInstance
[TPS_SYST_01007]	Definition of CommunicationController
[TPS_SYST_01008]	Assign CommunicationController to the AUTOSAR Communication Peripheral
[TPS_SYST_01009]	Definition of CommunicationConnector
[TPS_SYST_01010]	Definition of CommunicationCluster
[TPS_SYST_01011]	Definition of PhysicalChannel
[TPS_SYST_01012]	Different Properties of LinMaster and LinSlave
[TPS_SYST_01013]	EcuInstance stands for its own
[TPS_SYST_01014]	Semantics of CommunicationControllerMapping
[TPS_SYST_01015]	Semantics of HwPortMapping
[TPS_SYST_01016]	System Extract, Ecu System Description and Ecu Extract may have ports
[TPS_SYST_01017]	The role of the top-level software composition
[TPS_SYST_01019]	Mapping of topology elements to elements of the ECU Resource Template
[TPS_SYST_01020]	Unconditional mapping of atomic Software Components
[TPS_SYST_01021]	Mapping of CompositionSwComponentType
[TPS_SYST_01022]	Prototype of a ParameterSwComponentType can be mapped to more than one ECU
[TPS_SYST_01023]	Prototype of an ServiceProxySwComponentType can be mapped to more than one ECU
[TPS_SYST_01024]	Component Clustering
[TPS_SYST_01025]	Clustering of Compositions
[TPS_SYST_01026]	Separation of Compositions
[TPS_SYST_01027]	Mapping of specific SW components to dedicated Ecus
[TPS_SYST_01028]	Task of the System Generator
[TPS_SYST_01029]	Mapping of specific SW components to exclusive Ecus
[TPS_SYST_01030]	Representation of VariableDataPrototypes and ClientServerOperations in System Description
[TPS_SYST_01032]	Independence of SystemSignals from CommunicationClusters
[TPS_SYST_01033]	DataMapping and SwConnector
[TPS_SYST_01034]	Data Mappings can be applied to compositions and atomic software components
[TPS_SYST_01035]	Transformation of Data Mappings during flattening
[TPS_SYST_01036]	No additional Data Mappings in composition substructure
[TPS_SYST_01037]	primitive Data Mapping of UINT8-Arrays
[TPS_SYST_01038]	Mapping of primitive arguments
[TPS_SYST_01039]	primitive Argument Mapping of UINT8-Arrays
[TPS_SYST_01040]	Mapping of composite arguments
[TPS_SYST_01041]	CommonSignalPath definition
[TPS_SYST_01042]	ForbiddenSignalPath definition
[TPS_SYST_01043]	PermissibleSignalPath definition

[TPS_SYST_01044]	SeparateSignalPath definition
[TPS_SYST_01045]	Component Separation
[TPS_SYST_01046]	ShortNames of LinSlaveConfig and LinSlave
[TPS_SYST_01048]	Handling of large IPdus
[TPS_SYST_01049]	Handling of IPdus with dynamic signals
[TPS_SYST_01050]	SystemSignal in the System Extract and ECU Extract
[TPS_SYST_01051]	Handling of DcmIPdus
[TPS_SYST_01052]	Routing of UserDefinedPdu s, NmPdu s, NPdu s
[TPS_SYST_01053]	Low-level routing of NPdu s
[TPS_SYST_01054]	Routing of DcmIPdu s
[TPS_SYST_01055]	Routing of ISignalIPdu s that are part of a MultiplexedIPdu
[TPS_SYST_01056]	Routing of ISignalIPdu s, UserDefinedIPdu s, MultiplexedIPdu s
[TPS_SYST_01057]	Routing of NmPdu s
[TPS_SYST_01058]	Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content
[TPS_SYST_01059]	Relationship between FrameTriggering and CommConnectorPort
[TPS_SYST_01060]	Relationship between PduTriggering and CommConnectorPort
[TPS_SYST_01061]	Relationship between ISignalTriggering and CommConnectorPort
[TPS_SYST_01062]	Network representation of an ISignal
[TPS_SYST_01063]	Context of network representation of an ISignal
[TPS_SYST_01064]	Transmit/Receive Semantics of Pdu Pools
[TPS_SYST_01065]	Mapping onto the of ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01068]	Bit Counting in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01070]	E2E Protection of ISignalGroups
[TPS_SYST_01071]	E2E Protection of several ISignalGroups in one ISignalIPdu
[TPS_SYST_01072]	Offset attributes of EndToEndDescription
[TPS_SYST_01073]	E2E Protection via COM Callouts
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_01075]	Signal content evaluation via TransmissionModeCondition
[TPS_SYST_01076]	Mode evaluation via modeDrivenTrueCondition
[TPS_SYST_01077]	Mapping of Com Transmission Modes to System Template elements
[TPS_SYST_01078]	Dynamic Part of a MultiplexedIPdu
[TPS_SYST_01079]	Static Part of a MultiplexedIPdu
[TPS_SYST_01080]	Sending or receiving of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01081]	Gatewaying of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01082]	Receiving and gatewaying of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01083]	A Frame represents a general design object that is used to describe the layout of the included Pdu s as a reusable asset.
[TPS_SYST_01084]	FrameTriggering
[TPS_SYST_01085]	Transmission of a Frame multiple times within one communication cycle
[TPS_SYST_01086]	Number of Ethernet channels
[TPS_SYST_01087]	Role of SystemSignal in inter-ECU client server communication with clients located on the same ECU
[TPS_SYST_01088]	NetworkEndpoint priority
[TPS_SYST_01089]	ApplicationEndpoint priority
[TPS_SYST_01090]	valid NetworkEndpoint
[TPS_SYST_01091]	Definition of SoAdConfig
[TPS_SYST_01092]	Transmission of multiple Pdu s over the same SocketConnection

[TPS_SYST_01093]	Activation/Deactivation of SoAdRoutingGroups
[TPS_SYST_01094]	allowed key/value CapabilityRecord combinations
[TPS_SYST_01095]	tagged VLANs
[TPS_SYST_01096]	untagged VLANs
[TPS_SYST_01097]	Assignment of CouplingPorts to a VLAN
[TPS_SYST_01098]	Assignment of CouplingPorts to an “untagged” VLAN
[TPS_SYST_01099]	Context of TpConfig
[TPS_SYST_01100]	TP routing using the same transport protocol
[TPS_SYST_01101]	TP routing using different transport protocols
[TPS_SYST_01102]	FlexrayTpConnectionControl reuse
[TPS_SYST_01103]	FlexrayTpConnection shall specify one txPduPool
[TPS_SYST_01104]	FlexrayTpConnection with several receivers
[TPS_SYST_01105]	CanTpConnection with several receivers
[TPS_SYST_01106]	Usage of additional directPdu in case of variable length sdu
[TPS_SYST_01107]	Definition of NmCoordinator
[TPS_SYST_01108]	ProvidedServiceInstance priority
[TPS_SYST_01109]	RTE fan-out support
[TPS_SYST_01110]	Com Signal Gateway fan-out support
[TPS_SYST_01111]	Pdu Router fan-out support
[TPS_SYST_01112]	FlexRay dual channel Pdu Router interaction
[TPS_SYST_01113]	FlexRay Interface fan-out support
[TPS_SYST_01114]	Frame fan-out support
[TPS_SYST_01115]	CDD communication support
[TPS_SYST_01116]	Frame Mapping is not supported by the AUTOSAR BSW
[TPS_SYST_01117]	Pdu Gateway support
[TPS_SYST_01118]	Support of Multicast Pdu routing
[TPS_SYST_01119]	Signal Gateway support
[TPS_SYST_01120]	Precedence of ISignalMappings
[TPS_SYST_01121]	Support of Multicast signal routing
[TPS_SYST_01122]	partial routing between ISignalGroups
[TPS_SYST_01123]	System Extract may cover one or many EcuInstances
[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in
[TPS_SYST_01126]	Resource Consumption for RTE and basic software
[TPS_SYST_01127]	CDD Topology support
[TPS_SYST_01128]	Communication over FlexRay
[TPS_SYST_01129]	Communication over LIN
[TPS_SYST_01130]	Communication over CAN
[TPS_SYST_01131]	TCP/IP and UDP/IP communication over Ethernet
[TPS_SYST_01132]	Communication over SAE J1939
[TPS_SYST_01133]	Partial Network Clusters
[TPS_SYST_01134]	Abstract System Description
[TPS_SYST_01135]	Refactoring of an Abstract System Description into a project specific technical view of the software architecture
[TPS_SYST_01136]	ViewMapSet and ViewMap are used to trace the transformations between different models
[TPS_SYST_01137]	Several DataMappings may be defined for the same SystemSignal
[TPS_SYST_01138]	Low-level routing of XcpPdu s
[TPS_SYST_01139]	Ecu Extract covers exactly one EcuInstance
[TPS_SYST_01140]	Ecu Extract contains only SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype
[TPS_SYST_01141]	Derivation of ComSignalType
[TPS_SYST_01142]	Rules for the creation of Triggerings and Ports on the sender side

[TPS_SYST_01143]	DataMapping on the sender side for elements of a composite data type
[TPS_SYST_01144]	Physical properties of a System Signal
[TPS_SYST_01145]	PortInterfaceMappings in the ECU Extract
[TPS_SYST_01146]	Generic CanTpConnections
[TPS_SYST_01147]	Generic J1939TpConnections
[TPS_SYST_01148]	Mapping of IN and INOUT ArgumentDataPrototypes to callSignals
[TPS_SYST_01149]	Mapping of OUT and INOUT ArgumentDataPrototypes to returnSignals
[TPS_SYST_01150]	Mapping of returnSignal and callSignal to COM Signal
[TPS_SYST_01151]	DataMapping reference to an EventHandler
[TPS_SYST_01152]	DataMapping reference to a ConsumedEventGroup
[TPS_SYST_01153]	Atomic transport of SystemSignalGroups
[TPS_SYST_01154]	CAN Controller support of CAN FD frames
[TPS_SYST_03000]	Co-existing System with category SYSTEM_DESCRIPTION and System with category SYSTEM_EXTRACT
[TPS_SYST_05000]	System Description doesn't use a complete Software Component Description
[TPS_SYST_05001]	Send a Trigger across a network
[TPS_SYST_05002]	The value of startPosition is irrelevant

Table D.7: Added Specification Items in 4.1.1

D.4.5 Deleted Constraints in R4.1.1

[constr_3016]	Number of Ethernet channels

Table D.8: Deleted Constraints in R4.1.1

D.4.6 Deleted Specification Items in R4.1.1

N/A

D.5 Constraint and Specification Item History of this document according to AUTOSAR R4.1.2

D.5.1 Changed Specification Items in R4.1.2

Number	Heading
[TPS_SYST_01052]	Routing of UserDefinedPduS, NmPduS, NPduS, GeneralPurposePduS
[TPS_SYST_01056]	Routing of ISignalIPduS, UserDefinedIPduS, MultiplexedIPduS, GeneralPurposeIPduS
[TPS_SYST_01138]	Low-level routing of XcpPduS

Table D.9: Added Specification Items in 4.1.2

D.5.2 Added Specification Items in R4.1.2

Number	Heading
[TPS_SYST_02001]	<code>networkRepresentationProps</code> are mandatory in case the <code>dataTypePolicy</code> is set to <code>override</code> or <code>legacy</code>
[TPS_SYST_02002]	<code>SoAdRoutingGroup</code> for Services with Methods
[TPS_SYST_02003]	<code>SoAdRoutingGroups</code> for Services with event groups
[TPS_SYST_02004]	<code>SoAdRoutingGroups</code> for Services with event groups that contain triggered events
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus
[TPS_SYST_02006]	Usage of <code>networkRepresentationFromComSpec</code>
[TPS_SYST_02007]	Usage of <code>SocketConnection</code> attributes in the unicast server view
[TPS_SYST_02008]	Usage of <code>SocketConnection</code> attributes in the unicast client view
[TPS_SYST_02009]	Usage of <code>SocketConnection</code> attributes in the multicast server view
[TPS_SYST_02010]	Usage of <code>SocketConnection</code> attributes in the multicast client view

Table D.10: Added Specification Items in 4.1.2

D.5.3 Added Constraints in R4.1.2

Number	Heading
[constr_3069]	Allowed <code>CanNmCluster.nmNidPosition</code> values
[constr_3070]	Allowed <code>CanNmCluster.nmCbvPosition</code> values
[constr_3071]	<code>CanNmCluster.nmCbvPosition</code> and <code>CanNmCluster.nmNidPosition</code> shall never have the same value
[constr_3073]	<code>nmVoteInformation</code> only valid for <code>FrNm</code>
[constr_3074]	No <code>TransmissionAcknowledgementRequest</code> for multiple senders
[constr_3078]	Allowed <code>UdpNmCluster.nmNidPosition</code> values
[constr_3079]	Allowed <code>UdpNmCluster.nmCbvPosition</code> values
[constr_3080]	<code>UdpNmCluster.nmCbvPosition</code> and <code>UdpNmCluster.nmNidPosition</code> shall never have the same value
[constr_3081]	Value of category in <code>GeneralPurposePdu</code>
[constr_3082]	Value of category in <code>GeneralPurposeIPdu</code>
[constr_3083]	Exactly one <code>AtomicSwComponentType</code> on an <code>EcuInstance</code> may use <code>GeneralCallbackEventDataChanged</code> / <code>GeneralCallbackEventStatusChange</code>
[constr_3084]	Service port in the role <code>PowerTakeOff</code>
[constr_3085]	Service port in the role <code>CallbackDCMRequestServices</code>

Table D.11: Added Constraints in R4.1.2

D.5.4 Changed Constraints in R4.1.2

Number	Heading
[constr_2025]	Uniqueness of <code>symbol</code> attributes

Table D.12: Changed Constraints in R4.1.2

D.5.5 Deleted Constraints in R4.1.2

[constr_3066]	Restriction of <code>SenderComSpecs</code> that refer to <code>dataElements</code> mapped to the same <code>SystemSignal</code>

Table D.13: Deleted Constraints in R4.1.2

D.6 Constraint and Specification Item History of this document according to AUTOSAR R4.1.3

D.6.1 Changed Specification Items in R4.1.3

N/A

D.6.2 Added Specification Items in R4.1.3

Number	Heading
[TPS_SYST_01155]	Routing of ISignalGroups
[TPS_SYST_01156]	Definition of ISignalTriggerings is allowed for ISignalGroups and for GroupSignals
[TPS_SYST_01157]	Allowed usage of attributes for ISignals , ISignalGroups and GroupSignals
[TPS_SYST_02011]	initValues of receivers that are mapped to the same Ecu
[TPS_SYST_02012]	initValue and invalidValue represent internal values

Table D.14: Added Specification Items in 4.1.3

D.6.3 Deleted Specification Items in R4.1.3

Number	Heading
[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in

Table D.15: Deleted Specification Items in 4.1.3

D.6.4 Added Constraints in R4.1.3

[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3087]	DataMapping to PRPortPrototype
[constr_3088]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3090]	TpSdu transmission on a PhysicalChannel
[constr_3094]	Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals

Table D.16: Added Constraints in R4.1.3

D.6.5 Changed Constraints in R4.1.3

[constr_3051]	Restriction of ISignalMapping references

Table D.17: Changed Constraints in R4.1.3

D.6.6 Deleted Constraints in R4.1.3

[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_1206]	DataMapping to PRPortPrototype
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description

Table D.18: Deleted Constraints in R4.1.3

D.7 Constraint and Specification Item History of this document according to AUTOSAR R4.2.1

D.7.1 Added Traceables in 4.2.1

Id	Heading
[TPS_SYST_02013]	Usage of dataFilters on GroupSignals on receiver side
[TPS_SYST_02014]	ConsumedEventGroup priority
[TPS_SYST_02015]	LdCom: only one ISignal mapped to the ISignalIPdu
[TPS_SYST_02016]	LdCom: only Transformer output and UINT8_N or UINT8_DYN supported
[TPS_SYST_02017]	LdCom: Opaque ISignalToIPduMapping.packingByteOrder
[TPS_SYST_02018]	LdCom: ISignalToIPduMapping.startPosition shall be 0
[TPS_SYST_02019]	LdCom: ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition
[TPS_SYST_02020]	LdCom: No IPduTiming.minimumDelay defined
[TPS_SYST_02021]	LdCom: ISignalToIPduMapping.updateIndicationBitPosition shall not be defined
[TPS_SYST_02022]	LdCom: Only the transmissionModeTrueTiming defined
[TPS_SYST_02023]	LdCom: DataFilter "always" if TransmissionModeCondition defined
[TPS_SYST_02024]	LdCom: No ModeDrivenTransmissionModeCondition defined
[TPS_SYST_02025]	LdCom: Only EventControlledTiming defined
[TPS_SYST_02026]	LdCom: Only EventControlledTiming with no repetition defined
[TPS_SYST_02027]	LdCom: No ISignalPort.timeout reception timeout defined
[TPS_SYST_02028]	LdCom: No ISignalPort.dataFilter defined
[TPS_SYST_02029]	Multiple ParameterDataPrototype instances in an EcuExtract
[TPS_SYST_02030]	The DataTransformationSet contains all transformer chains
[TPS_SYST_02031]	A transformer is represented by a TransformationTechnology
[TPS_SYST_02032]	Transformer chains are ordered list of transformers
[TPS_SYST_02033]	Order of the transformers in the configuration represents the order on the sending side
[TPS_SYST_02034]	Order of the transformers on the receiving side is the reverse of the sending side

[TPS_SYST_02035]	<code>protocol</code> contains the human readable protocol identifier
[TPS_SYST_02036]	<code>version</code> contains the version of the <code>protocol</code>
[TPS_SYST_02037]	The attribute <code>needsOriginalData</code> configures a transformer's access to the original data
[TPS_SYST_02038]	Specification of transformer class
[TPS_SYST_02039]	Specification of transformer specific properties
[TPS_SYST_02040]	Specification of transformer buffer handling
[TPS_SYST_02041]	In-place buffer handling of transformers
[TPS_SYST_02042]	Header length to be considered by transformers
[TPS_SYST_02043]	Buffer computation of transformer
[TPS_SYST_02044]	Buffer computation of transformer
[TPS_SYST_02045]	SOME/IP Transformer configuration
[TPS_SYST_02046]	E2E Transformer configuration
[TPS_SYST_02047]	Custom transformer configuration
[TPS_SYST_02048]	<code>ISignal</code> specific transformation configuration
[TPS_SYST_02049]	Transformer specific <code>TransformationISignalProps</code>
[TPS_SYST_02050]	<code>ISignal</code> specific configuration of the SOME/IP Transformer
[TPS_SYST_02051]	<code>ISignal</code> specific configuration of the E2E Transformer
[TPS_SYST_02052]	<code>ISignal</code> specific configuration of custom transformers
[TPS_SYST_02053]	A reference from <code>ISignal</code> to <code>DataTransformation</code> in the role <code>data-Transformation</code> enables data transformation
[TPS_SYST_02054]	Definition of data which shall be transformed
[TPS_SYST_02055]	Alignment of SOME/IP
[TPS_SYST_02056]	Byte Order of SOME/IP
[TPS_SYST_02057]	Interface Version of SOME/IP
[TPS_SYST_02058]	Usage of COM Based Transformer
[TPS_SYST_02059]	Routing of <code>SecuredIPdus</code>
[TPS_SYST_02060]	<code>SecuredIPdus</code>
[TPS_SYST_02061]	Routing of <code>IPdus</code> that are part of a <code>ContainerIPdu</code>
[TPS_SYST_02062]	Allowed <code>ContainedIPduProps.headerIdLongHeader</code> and <code>ContainedIPduProps.headerIdShortHeader</code> values
[TPS_SYST_02063]	Byte order of <code>ContainerIPdu</code> header information
[TPS_SYST_02064]	Reception acceptance of contained <code>IPdus</code>
[TPS_SYST_02065]	Contained <code>IPdu</code> specific transmission timeout
[TPS_SYST_02066]	<code>ContainerIPdu.thresholdSize</code>
[TPS_SYST_02067]	E2E profile
[TPS_SYST_02068]	E2E header field representation in an <code>ISignalGroup</code>
[TPS_SYST_02069]	Recommended configuration settings for E2E Profile 1 configuration setting C
[TPS_SYST_02070]	Recommended configuration settings for E2E Profile 4 configuration setting A
[TPS_SYST_02071]	Recommended configuration settings for E2E Profile 4 configuration setting B
[TPS_SYST_02072]	<code>profileName</code> of <code>EndToEndTransformationDescription</code>
[TPS_SYST_02073]	<code>EndToEndTransformationDescription.profileName</code>
[TPS_SYST_02074]	Precedence of transformer configuration settings
[TPS_SYST_02075]	Mandatory attributes in transformer configuration elements
[TPS_SYST_03001]	LdCom: <code>ISignalIPdu</code> not part of any <code>ISignalIPduGroup</code>
[TPS_SYST_03002]	Keep behavior of DHCP clients
[TPS_SYST_03003]	Ethernet priority regeneration
[TPS_SYST_03004]	VLAN specific sending behavior
[TPS_SYST_03005]	VLAN re-tagging
[TPS_SYST_03006]	Ethernet switch egress port setup

[TPS_SYST_03007]	Ethernet port scheduler algorithm
[TPS_SYST_03008]	Ethernet port scheduler priority
[TPS_SYST_03009]	Ethernet port shaper <code>idleSlope</code>
[TPS_SYST_03010]	Ethernet switch packet to traffic class assignment
[TPS_SYST_03011]	Ethernet switch traffic class to FIFO assignment
[TPS_SYST_03013]	Semi-static DHCP server configuration
[TPS_SYST_03014]	Transmission triggering by the first contained <code>IPdu</code> put into a <code>Container-IPdu</code>
[TPS_SYST_05003]	Usage of <code>DiagnosticConnection</code> in combination with a TP
[TPS_SYST_05004]	Usage of <code>DiagnosticConnection</code> in combination with UUDT
[TPS_SYST_05005]	Relation of <code>GlobalTimeDomain</code> to <code>CommunicationCluster</code>
[TPS_SYST_05006]	Chaining of <code>GlobalTimeDomains</code>
[TPS_SYST_05007]	separation of roles within a <code>GlobalTimeDomain</code>
[TPS_SYST_05008]	Semantics of a <code>GlobalTimeGateway</code>
[TPS_SYST_05009]	<code>Pdu</code> for transmitting global time information
[TPS_SYST_05010]	<code>Pdu</code> is not required on Ethernet
[TPS_SYST_05011]	Ownership of <code>GlobalTimeGateway</code>
[TPS_SYST_05013]	Semantics of <code>GlobalTimeMaster.isSystemWideGlobalTimeMaster</code>
[TPS_SYST_05014]	<code>GlobalTimeMaster.isSystemWideGlobalTimeMaster</code>
[TPS_SYST_05015]	Naming conventions

Table D.19: Added Traceables in 4.2.1

D.7.2 Changed Traceables in 4.2.1

Id	Heading
[TPS_SYST_01024]	Component Clustering
[TPS_SYST_01025]	Clustering of Compositions
[TPS_SYST_01026]	Separation of Compositions
[TPS_SYST_01045]	Component Separation
[TPS_SYST_01056]	Routing of <code>ISignalIPdus</code> , <code>UserDefinedIPdus</code> , <code>MultiplexedIPdus</code> , <code>GeneralPurposeIPdus</code> , <code>ContainerIPdus</code>
[TPS_SYST_01057]	Routing of <code>NmPdus</code>
[TPS_SYST_01088]	<code>NetworkEndpoint</code> priority
[TPS_SYST_01089]	<code>ApplicationEndpoint</code> priority
[TPS_SYST_01106]	Usage of additional <code>directPdu</code> in case of variable length <code>sdu</code>
[TPS_SYST_01108]	<code>ProvidedServiceInstance</code> priority
[TPS_SYST_01138]	Low-level routing of <code>XcpPdus</code>
[TPS_SYST_01157]	Allowed usage of attributes for <code>ISignals</code> , <code>ISignalGroups</code> and <code>GroupSignals</code>
[TPS_SYST_02005]	Low-level routing of <code>J1939DcmIPdus</code>

Table D.20: Changed Traceables in 4.2.1

D.7.3 Deleted Traceables in 4.2.1

Id	Heading
[TPS_SYST_01038]	Mapping of primitive arguments
[TPS_SYST_01039]	primitive Argument Mapping of UIN8-Arrays
[TPS_SYST_01040]	Mapping of composite arguments
[TPS_SYST_01051]	Handling of <code>DcmIPdus</code>

Table D.21: Deleted Traceables in 4.2.1

D.7.4 Added Constraints in 4.2.1

Id	Heading
[constr_1367]	<code>periodicResponseUdt.periodicResponseUdt</code> shall only refer to a <code>DcmIPdu</code>
[constr_1368]	Limitation of the target of references from <code>DiagnosticConnection</code>
[constr_1369]	<code>CommunicationConnectors</code> shall be attached to the same <code>Communication-Cluster</code>
[constr_1370]	Consistency of <code>GlobalTimeDomain</code>
[constr_1371]	Consistency of attribute <code>host</code>
[constr_1372]	Consistency of attribute <code>globalTimePdu</code>
[constr_1373]	<code>GlobalTimeMaster</code> with attribute <code>isSystemWideGlobalTimeMaster</code> set to TRUE
[constr_1374]	Only fan-out possible for <code>GlobalTimeGateway</code>
[constr_3095]	<code>canControllerFdAttributes</code> and <code>canControllerFdRequirements</code> are mutually exclusive.
[constr_3096]	Allowed values for <code>diagnosticMessageType</code>
[constr_3097]	Overlapping of segments of one <code>MultiplexedIPdu</code> is not allowed
[constr_3098]	Defined segments of one <code>MultiplexedIPdu</code> shall not exceed the length of the <code>MultiplexedIPdu</code>
[constr_3099]	Defined segments in a <code>DynamicPart</code> shall not exceed the length of any <code>Dynamic-PartAlternative.ipdu</code>
[constr_3100]	Defined segments in a <code>StaticPart</code> shall not exceed the length of the <code>StaticPart.ipdu</code>
[constr_3101]	Signal representation of selector field for <code>DynamicPartAlternative</code>
[constr_3102]	Restriction on usage of <code>J1939NodeName</code> attributes
[constr_3103]	Range of <code>ecuInstance</code>
[constr_3104]	Range of <code>function</code>
[constr_3105]	Range of <code>functionInstance</code>
[constr_3106]	Range of <code>identityNumber</code>
[constr_3107]	Range of <code>industryGroup</code>
[constr_3108]	Range of <code>manufacturerCode</code>
[constr_3109]	Range of <code>vehicleSystem</code>
[constr_3110]	Range of <code>vehicleSystemInstance</code>
[constr_3111]	<code>returnSignal</code> in <code>ClientServerToSignalMapping</code> is mandatory
[constr_3112]	Invalidation support for partial mapping of a data element typed by composite data type
[constr_3113]	<code>EthernetFrame</code> shall not have a <code>PduToFrameMapping</code>
[constr_3114]	<code>FlatInstanceDescriptors</code> pointing to the same <code>ParameterDataPrototype</code> shall have different <code>postBuildVariantConditions</code>
[constr_3115]	<code>FlatInstanceDescriptors</code> pointing to the same <code>ParameterDataPrototype</code> instance
[constr_3116]	Overlap of <code>ClientIdRanges</code> in the context of the enclosing System
[constr_3117]	Allowed value of attribute <code>clientId</code>
[constr_3118]	Valid reference target for <code>ClientIdDefinition.clientServerOperation.contextPort</code>
[constr_3121]	The length of transformer chains is limited to 255 transformers
[constr_3122]	At most one transformer of each transformer class inside a transformer chain
[constr_3123]	Serializer transformer shall be the first in a chain
[constr_3124]	Applicability of <code>needsOriginalData</code>

[constr_3125]	Value of attribute <code>inPlace</code> for the first transformer in a chain
[constr_3126]	<code>headerLength</code> shall be less or equal output buffer size
[constr_3127]	Certain <code>ISignals</code> always need a reference to <code>DataTransformation</code>
[constr_3128]	SOME/IP transformer configuration
[constr_3129]	Byte Order of SOME/IP transformer
[constr_3130]	Range of Interface Version
[constr_3131]	Required first data transformation for <code>comBasedSignalGroupTransformation</code>
[constr_3132]	Required COM Based Transformation for <code>comBasedSignalGroupTransformation</code>
[constr_3133]	<code>physicalLayerType</code> of connected <code>CouplingPorts</code>
[constr_3134]	The connection of two <code>CouplingPorts</code> with <code>connectionNegotiationBehavior</code> set to <code>master</code> is forbidden
[constr_3135]	The connection of two <code>CouplingPorts</code> with <code>connectionNegotiationBehavior</code> set to <code>slave</code> is forbidden
[constr_3136]	Allowed payload of <code>SecuredIPdus</code>
[constr_3137]	<code>IPduPort.rxSecurityVerification</code> is configurable on the receiver side
[constr_3138]	<code>IPduPort.rxSecurityVerification</code> validness
[constr_3139]	Usage of <code>IPduPort.rxSecurityVerification</code>
[constr_3140]	No <code>ByteOrderEnum.opaque</code> allowed for <code>System.containerIPduHeaderByteOrder</code>
[constr_3141]	Only <code>IPdus</code> shall be part of a <code>ContainerIPdu</code>
[constr_3142]	Mandatory <code>headerIdLongHeader</code> for <code>longHeader</code>
[constr_3143]	Mandatory <code>headerIdShortHeader</code> for <code>shortHeader</code>
[constr_3144]	Mandatory <code>IPdu.containedIPduProps</code> for contained <code>IPdus</code>
[constr_3146]	Partial Networking timing constraint
[constr_3148]	<code>executeDespiteDataUnavailability</code> setting in case an E2E Transformer is used
[constr_3149]	<code>TransformationTechnology</code> settings for E2E Transformer
[constr_3150]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_01 in case it is 0
[constr_3151]	<code>BufferProperties.headerLength</code> settings for an E2E transformer used in combination with a SOME/IP transformer
[constr_3152]	<code>BufferProperties.headerLength</code> settings for an E2E transformer used in combination with a COM Based transformer
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3154]	<code>BufferProperties.bufferComputation</code> setting for an E2E transformer
[constr_3155]	Allowed values for <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code>
[constr_3156]	Allowed values for <code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_01
[constr_3157]	Allowed values for <code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_01 in case <code>dataIdMode</code> is set to <code>lower12Bit</code>
[constr_3158]	Allowed values for <code>EndToEndTransformationDescription.maxDeltaCounter</code> in PROFILE_01
[constr_3159]	Allowed values for <code>EndToEndTransformationDescription.maxDeltaCounter</code> in PROFILE_04
[constr_3160]	<code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_02
[constr_3161]	<code>EndToEndTransformationISignalProps.dataLength</code> in PROFILE_01, PROFILE_02, PROFILE_05
[constr_3162]	<code>EndToEndTransformationISignalProps.minDataLength</code> and <code>EndToEndTransformationISignalProps.maxDataLength</code> in PROFILE_01, PROFILE_02, PROFILE_05

[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04 and PROFILE_06
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04 and PROFILE_06
[constr_3165]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01
[constr_3166]	EndToEndTransformationDescription.upperHeaderBitsToShift in PROFILE_02
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3169]	Attribute multiplicities and values in PROFILE_02
[constr_3171]	Value of EndToEndTransformationISignalProps.dataId shall be unique in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3172]	Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_01
[constr_3173]	Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_02
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3176]	Value range of windowSize
[constr_3177]	Dependency between maxErrorStateValid, maxErrorStateInit and maxErrorStateInvalid
[constr_3178]	Dependency between minOkStateValid, minOkStateInit and minOkStateInvalid
[constr_3179]	Dependency between minOkStateInit, maxErrorStateInit and windowSizeStateInit
[constr_3180]	Dependency between minOkStateValid, maxErrorStateValid and windowSizeStateValid
[constr_3181]	Dependency between minOkStateInvalid, maxErrorStateInvalid and windowSizeStateInvalid
[constr_3182]	Restriction on TransformationTechnology.transformationDescriptionVariationPoint
[constr_3183]	ISignalGroup with transformationISignalProps
[constr_3184]	Only one EndToEndTransformationISignalProps.dataId element in PROFILE_01
[constr_3185]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_01
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3187]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_01
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3189]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_01
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3191]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_01 and dataIdMode equal to lower12Bit
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06 or dataIdMode different from lower12Bit
[constr_3193]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_01

[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in Profiles different from PROFILE_01
[constr_3195]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02
[constr_3196]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_05
[constr_3197]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_06
[constr_3515]	Fully filled EthernetPriorityRegeneration table
[constr_3516]	limitation of Pdu.length for CAN L-PDUs
[constr_3517]	Consistent setting of ContainedIPduProps.collectionSemantics in the context of one ContainerIPdu
[constr_3518]	Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Table D.22: Added Constraints in 4.2.1

D.7.5 Changed Constraints in 4.2.1

Id	Heading
[constr_3010]	ISignalIPdu length shall not be exceeded
[constr_3011]	Overlapping of updateIndicationBits of ISignals is prohibited
[constr_3028]	FibexElements
[constr_3037]	maximum Frame.frameLength for CAN and LIN
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup

Table D.23: Changed Constraints in 4.2.1

D.7.6 Deleted Constraints in 4.2.1

Id	Heading
[constr_1208]	Existence of the attribute DataMapping.communicationDirection in the context of a ClientServerInterface
[constr_3001]	valid ClientServerToSignalGroupMappings
[constr_3017]	Length of multiplexed Pdu shall not be exceeded.
[constr_3026]	valid EmptySignalMappings
[constr_3033]	Criteria for primitive argument mapping
[constr_3056]	pduLength of the NmPdu

Table D.24: Deleted Constraints in 4.2.1

D.8 Constraint and Specification Item History of this document according to AUTOSAR R4.2.2

D.8.1 Added Traceables in 4.2.2

Id	Heading
----	---------

[TPS_SYST_02076]	networkRepresentationProps in case the dataTypePolicy is set to transformingISignal
[TPS_SYST_02077]	Subscribers of a LinEventTriggeredFrame
[TPS_SYST_02078]	LinUnconditionalFrames associated with a LinEventTriggeredFrame
[TPS_SYST_02079]	Identification of ImplementationDataType for a given ISignal in an Ecu Extract
[TPS_SYST_02080]	Message type of SOME/IP
[TPS_SYST_02081]	PduTriggering that is used for ClientServer Communication
[TPS_SYST_02082]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE or BOOLEAN and a DataTypeMap exists
[TPS_SYST_02083]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and a DataTypeMap exists
[TPS_SYST_02084]	SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and a DataTypeMap exists
[TPS_SYST_02085]	SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category ARRAY
[TPS_SYST_02086]	SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category VALUE or TYPE_REFERENCE
[TPS_SYST_02087]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category BOOLEAN and no DataTypeMap exists
[TPS_SYST_02088]	SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and no DataTypeMap exists
[TPS_SYST_02089]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and no DataTypeMap exists
[TPS_SYST_02090]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE and no DataTypeMap exists
[TPS_SYST_02091]	Routing of GeneralPurposePdus with category SD and GeneralPurposePdus with category DoIP
[TPS_SYST_02092]	Size of Fixed-size Array Length Fields
[TPS_SYST_02093]	Size of Structure Length Fields
[TPS_SYST_02094]	Size of Union Length Fields
[TPS_SYST_03015]	Offset time domain requires synchronized time domain

Table D.25: Added Traceables in 4.2.2

D.8.2 Changed Traceables in 4.2.2

Id	Heading
[TPS_SYST_01003]	Standardized System Category Definitions
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus with category GLOBAL_TIME
[TPS_SYST_01065]	Mapping onto the ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01077]	Mapping of Com Transmission Modes to System Template elements
[TPS_SYST_01113]	FlexRay Interface fan-out support

[TPS_SYST_01157]	Allowed usage of attributes for <code>ISignals</code> , <code>ISignalGroups</code> and <code>GroupSignals</code>
[TPS_SYST_02017]	LdCom: Opaque <code>ISignalToIPduMapping.packingByteOrder</code>
[TPS_SYST_02069]	Recommended configuration settings for E2E Profile 1 configuration setting C
[TPS_SYST_02070]	Recommended configuration settings for E2E Profile 4 configuration setting A
[TPS_SYST_02071]	Recommended configuration settings for E2E Profile 4 configuration setting B

Table D.26: Changed Traceables in 4.2.2

D.8.3 Deleted Traceables in 4.2.2

none

D.8.4 Added Constraints in 4.2.2

Id	Heading
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_1387]	Transmission of Variable-Size Array Data Types by means of a Transformer
[constr_3198]	Uniqueness of <code>PncMapping.shortLabel</code>
[constr_3199]	<code>ISignal</code> that has <code>dataTypePolicy</code> set to <code>transformingISignal</code> shall reference a <code>DataTransformation</code>
[constr_3201]	<code>eventGroupIdentifier</code> in <code>ConsumedEventGroups</code> that are referenced by the same <code>EventHandler</code>
[constr_3202]	<code>LinFrameTriggering</code> to <code>LinUnconditionalFrame</code> reference restriction in <code>LinEventTriggeredFrame</code> context
[constr_3203]	<code>LinFrameTriggering</code> to <code>LinSporadicFrame</code> reference restriction in <code>LinSporadicFrame</code> context
[constr_3204]	<code>LinUnconditionalFrames</code> associated with a <code>LinSporadicFrame</code>
[constr_3205]	Existence of <code>FramePort</code> for a <code>FrameTriggering</code> that references a <code>LinSporadicFrame</code>
[constr_3206]	Existence of <code>FramePort</code> for a <code>FrameTriggering</code> that references a <code>LinEventTriggeredFrame</code>
[constr_3207]	Assignment of <code>SocketConnectionIpduIdentifiers</code> used for ClientServer Communication to <code>SocketConnections</code>
[constr_3208]	<code>executeDespiteDataUnavailability</code> usage restriction
[constr_3209]	<code>CanFrameTriggerings</code> with identical PGN
[constr_3210]	<code>J1939TpPgs</code> with identical <code>pgn</code> value
[constr_3211]	<code>PduTriggerings</code> with <code>triggerIPduSendCondition</code>
[constr_3212]	Limitation of <code>DolpTpConnection.tpSdu</code>
[constr_3213]	<code>TransformationISignalProps.csErrorReaction</code> setting in case that the <code>serializer transformerClass</code> and Client/Server communication is used
[constr_3214]	<code>TransformationISignalProps.csErrorReaction</code> setting in case that a <code>transformerClass</code> different from <code>serializer</code> is used or the Client/Server communication is not used
[constr_3215]	<code>TransformationTechnology.version</code> and <code>TransformationTechnology.protocol</code> settings for request and response of a client/server communication
[constr_3216]	Usage of <code>SOMEIPTransformationISignalProps.sessionHandlingSR</code>
[constr_3218]	Range of Size of Fixed-size Array Length Fields

[constr_3219]	The existence of LinSlaves in the LinMaster EcuExtract
[constr_3220]	Range of Size of Structure Length Fields
[constr_3221]	Range of Size of Union Length Fields
[constr_3519]	Value of category of GlobalTimeDomain
[constr_3520]	Offset time domain shall be based on a synchronized time domain

Table D.27: Added Constraints in 4.2.2

D.8.5 Changed Constraints in 4.2.2

Id	Heading
[constr_1368]	Limitation of the target of references from DiagnosticConnection
[constr_1374]	Only fan-out possible for GlobalTimeGateway
[constr_3002]	valid swcToImplMapping
[constr_3003]	Number of CAN channels
[constr_3004]	Clustering and separation must be exclusive
[constr_3005]	valid EcuResourceEstimation
[constr_3006]	valid EcuMapping
[constr_3007]	selectorFieldCodes for dynamic part alternatives
[constr_3008]	EcuInstance subelements
[constr_3015]	Number of LIN channels
[constr_3018]	Number of FlexRay channels
[constr_3019]	In the flat ECU extract each required interface must be satisfied by connected provided interfaces
[constr_3020]	communicationDirection of containedIPduGroups
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3025]	Usage of NPdus in TpConnections
[constr_3027]	Existence of ecuExtractVersion
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3095]	canControllerFdAttributes and canControllerFdRequirements are mutually exclusive
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3179]	Dependency between minOkStateInit , maxErrorStateInit and windowSize
[constr_3180]	Dependency between minOkStateValid , maxErrorStateValid and windowSize
[constr_3181]	Dependency between minOkStateInvalid , maxErrorStateInvalid and windowSize
[constr_3501]	Role of SystemSignal in 1:n communication
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3518]	Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Table D.28: Changed Constraints in 4.2.2

D.8.6 Deleted Constraints in 4.2.2

Id	Heading
[constr_3131]	Required first data transformation for comBasedSignalGroupTransformation
[constr_3505]	Criteria for primitive Data Mapping

Table D.29: Deleted Constraints in 4.2.2

D.9 Constraint and Specification Item History of this document according to AUTOSAR R4.3.0

D.9.1 Added Traceables in 4.3.0

Id	Heading
[TPS_SYST_02095]	LinFrameTriggering.linChecksum for LinUnconditionalFrames
[TPS_SYST_02096]	Sending of ANY finds for minor version
[TPS_SYST_02097]	Basic definition of contained IPdu s
[TPS_SYST_02098]	Header id and header type of a contained IPdu
[TPS_SYST_02099]	Relation between ContainerIPdu and contained IPdu s on sender side
[TPS_SYST_02100]	Relation between ContainerIPdu and contained IPdu s on receiver side
[TPS_SYST_02101]	Usage of LinSlaveConfig in Ecu Extract
[TPS_SYST_02102]	FrameTriggering.pduTriggering references that shall be ignored
[TPS_SYST_02103]	Semantics of GlobalTimeDomain.domainId
[TPS_SYST_02104]	Triggerings on PhysicalChannel
[TPS_SYST_02105]	ISignalGroup and ISignal referenced from ISignalTriggering
[TPS_SYST_02106]	Rules for the creation of references to Ports (ecuCommPortInstance) with communicationDirection in on receiving Ecu
[TPS_SYST_02107]	Shared address space for J1939 routing relations
[TPS_SYST_02108]	Address proxying for J1939 routing relations
[TPS_SYST_02109]	Absence of J1939SharedAddressCluster.participatingJ1939Cluster to a J1939Cluster
[TPS_SYST_02110]	Default behavior for ISignal.iSignalType
[TPS_SYST_02111]	VariableDataPrototype in case ISignal.iSignalType is set to array
[TPS_SYST_02112]	Usage of EventHandler.applicationEndpoint reference
[TPS_SYST_02113]	Usage of ConsumedEventGroup.applicationEndpoint reference
[TPS_SYST_02114]	Mapping of SwComponentPrototypes onto SwcToEcuMapping targets
[TPS_SYST_02115]	Applicability of GlobalTimeDomain.globalTimeDomainProps
[TPS_SYST_02116]	Modeling of Service Discovery Pdu s
[TPS_SYST_02117]	Length of GeneralPurposePdu with category SD
[TPS_SYST_02118]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdu s with category SD
[TPS_SYST_02119]	SocketConnectionBundles for GeneralPurposePdu s with category SD
[TPS_SYST_02120]	runtimeIpAddressConfiguration and runtimePortConfiguration settings for SD SocketConnections
[TPS_SYST_02121]	Scope of DataPrototypeTransformationProps
[TPS_SYST_02123]	Size of a length field for a chosen fixed-size array
[TPS_SYST_02124]	Size of a length field for a chosen structure
[TPS_SYST_02125]	Size of a length field for a chosen union
[TPS_SYST_02126]	Alignment of a dynamic DataPrototype
[TPS_SYST_02127]	Usage of DataPrototypeTransformationProps in case of a VariableDataPrototype

[TPS_SYST_02128]	Usage of DataPrototypeTransformationProps in case of a ClientServerOperation
[TPS_SYST_02129]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype typed by an ApplicationDataType
[TPS_SYST_02130]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ApplicationDataType
[TPS_SYST_02131]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype typed by an ImplementationDataType
[TPS_SYST_02132]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ImplementationDataType
[TPS_SYST_02133]	BufferProperties.bufferComputation setting for a COM Based transformer
[TPS_SYST_02134]	Recommended configuration settings for E2E Profile 7 configuration setting A
[TPS_SYST_02135]	Recommended configuration settings for E2E Profile 7 configuration setting B
[TPS_SYST_02136]	Serialization based on the network representation
[TPS_SYST_02137]	Serialization based on the ImplementationDataTypes
[TPS_SYST_02138]	Definition of the network representation
[TPS_SYST_02139]	Applicability of the SwDataDefProps attributes for the network representation of the serialized data
[TPS_SYST_02140]	SocketConnectionBundle.udpChecksumHandling default value
[TPS_SYST_02141]	Semantics of udpChecksumHandling
[TPS_SYST_02142]	Reception of invalid checksum
[TPS_SYST_02143]	Support of Multisource Pdu routing
[TPS_SYST_02144]	ComTimeoutSubstitution does not apply for signal gateway operation
[TPS_SYST_02145]	Default behavior for not defined nmPncParticipation
[TPS_SYST_02146]	Explicit definition of pncVector at NmPdu
[TPS_SYST_02147]	Implicit definition of pncVector at NmPdu
[TPS_SYST_02148]	Meaning of useAsCryptographicIPdu that is not set or set to false
[TPS_SYST_02149]	Meaning of useAsCryptographicIPdu that is set to true
[TPS_SYST_02150]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that LdCom is used
[TPS_SYST_02151]	MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs if one SystemSignal per communication direction is used
[TPS_SYST_02152]	Security profile
[TPS_SYST_02153]	Standardized values for the attribute category of meta-class SecureCommunicationFreshnessProps
[TPS_SYST_02154]	Standardized values for the attribute category of meta-class SecureCommunicationAuthenticationProps
[TPS_SYST_02155]	Recommended configuration settings for E2E Profile 11 configuration setting C
[TPS_SYST_02156]	Length of GeneralPurposeIPdu with category SOMEIP_SEGMENTED_IPDU
[TPS_SYST_02157]	Default value for the attribute category of meta-class EthernetCommunicationConnector
[TPS_SYST_02158]	Default value for the attribute category of meta-class EthernetCommunicationController
[TPS_SYST_02159]	Default value for the attribute category of meta-class EthernetPhysicalChannel
[TPS_SYST_02160]	EthernetPhysicalChannels with different category values are not allowed within an EthernetCluster

[TPS_SYST_02161]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that Com is used
[TPS_SYST_03016]	Applicability of EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod
[TPS_SYST_03017]	Reference to CouplingPort in the context of a GlobalTimeDomain
[TPS_SYST_03018]	Aggregation of PNCs at the hostPort
[TPS_SYST_03019]	Modeling of CouplingPorts for managed CouplingElement
[TPS_SYST_03020]	Default value for CouplingPort.couplingPortRole if not defined
[TPS_SYST_03021]	Routing of GeneralPurposePdus with category GLOBAL_TIME

Table D.30: Added Traceables in 4.3.0

D.9.2 Changed Traceables in 4.3.0

Id	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01052]	Routing of UserDefinedPdus , NmPduS , NPdus
[TPS_SYST_01065]	Mapping onto the ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01118]	Support of Multicast Pdu routing
[TPS_SYST_01142]	Rules for the creation of references to Ports (ecuCommPortInstance) with communicationDirection out on sending Ecu
[TPS_SYST_02002]	SoAdRoutingGroup for Services with Methods
[TPS_SYST_02003]	SoAdRoutingGroups for Services with event groups
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02033]	Order of the transformerChain references in the configuration represents the order on the sending side
[TPS_SYST_02044]	Buffer computation of transformer
[TPS_SYST_02073]	EndToEndTransformationDescription.profileName
[TPS_SYST_02092]	Size of Fixed-size Array Length Fields
[TPS_SYST_02093]	Size of Structure Length Fields
[TPS_SYST_02094]	Size of Union Length Fields
[TPS_SYST_05009]	GlobalTimeDomain.globalTimePduTriggering for transmitting global time information
[TPS_SYST_05010]	GlobalTimeDomain.globalTimePduTriggering is not required on Ethernet

Table D.31: Changed Traceables in 4.3.0

D.9.3 Deleted Traceables in 4.3.0

Id	Heading
[TPS_SYST_01027]	Mapping of specific SW components to dedicated Ecus
[TPS_SYST_01028]	Task of the System Generator
[TPS_SYST_01029]	Mapping of specific SW components to exclusive Ecus
[TPS_SYST_01141]	Derivation of ComSignalType

Table D.32: Deleted Traceables in 4.3.0

D.9.4 Added Constraints in 4.3.0

Id	Heading
[constr_1441]	In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer
[constr_1463]	Applicable values for <code>J1939Cluster.networkId</code>
[constr_3222]	No <code>ByteOrderEnum.opaque</code> allowed for <code>PduToFrameMapping.packingByteOrder</code>
[constr_3223]	No <code>ByteOrderEnum.opaque</code> allowed for <code>MultiplexedIPdu.selectorFieldByteOrder</code>
[constr_3224]	No <code>ByteOrderEnum.opaque</code> allowed for <code>SegmentPosition.segmentByteOrder</code> .
[constr_3225]	<code>LinFrameTriggering.linChecksum</code> not allowed for <code>LinSporadicFrames</code>
[constr_3226]	<code>LinFrameTriggering.linChecksum</code> for <code>LinEventTriggeredFrames</code>
[constr_3227]	<code>NmNode.nmPassiveModeEnabled</code> setting
[constr_3229]	<code>SwComponentPrototype</code> mapped to an <code>ApplicationPartition</code> and <code>EcuInstance</code>
[constr_3230]	Usage of <code>SenderRecRecordElementMapping.applicationRecordElement</code>
[constr_3231]	Usage of <code>IndexedArrayElement.applicationArrayElement</code>
[constr_3232]	<code>ApplicationPartition</code> is allowed to be mapped to only one <code>EcuPartition</code>
[constr_3239]	Consistent mapping of software-component to <code>J1939NmNode</code>
[constr_3240]	Consistent mapping of <code>J1939ControllerApplication</code> to <code>EcuInstance</code>
[constr_3241]	Usage of <code>AssignFrameId.messageId</code>
[constr_3242]	Usage of <code>UnassignFrameId.messageId</code>
[constr_3243]	<code>FrameTriggering.pduTriggering</code> condition
[constr_3244]	Usage of <code>SenderRecRecordElementMapping.implementationRecordElement</code>
[constr_3245]	Usage of <code>IndexedArrayElement.implementationArrayElement</code>
[constr_3246]	<code>Frame.packingByteOrder</code> mix within a <code>Frame</code> is not allowed
[constr_3247]	Byte order mix within a <code>MultiplexedIPdu</code> is not allowed
[constr_3248]	Category of <code>HwElement</code> for <code>ECUMapping</code>
[constr_3249]	Category of <code>HwElement</code> for <code>SwcToEcuMapping</code>
[constr_3250]	<code>PduTriggering.iSignalTriggering</code> condition
[constr_3251]	Value of <code>GlobalTimeDomain.domainId</code> in subDomain chains
[constr_3252]	<code>ISignalTriggering.iSignalPort</code> reference condition
[constr_3253]	<code>PduTriggering.iPduPort</code> reference condition
[constr_3254]	<code>FrameTriggering.framePort</code> reference condition
[constr_3255]	<code>FrameTriggering.pduTriggering</code> reference condition with regard to the <code>PhysicalChannel</code>
[constr_3256]	<code>PduTriggering.iSignalTriggering</code> reference condition with regard to the <code>PhysicalChannel</code>
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3258]	Restriction on <code>ISignal.length</code> in case <code>iSignalType</code> is set to <code>array</code>
[constr_3259]	Allowed use of <code>SdServerConfig.capabilityRecord</code>
[constr_3260]	Allowed use of <code>SdClientConfig.capabilityRecord</code>
[constr_3261]	<code>GlobalTimeDomain.globalTimePduTriggering</code> category
[constr_3262]	<code>ConsumedEventGroup.eventGroupIdentifier</code> is mandatory
[constr_3263]	Restriction of usage of <code>SwcToEcuMapping</code> in a <code>System</code>
[constr_3264]	Server side <code>ClientServerToSignalMappings</code> in case of a n:1 inter-ECU client-server communication
[constr_3265]	<code>TransformationTechnology.hasInternalState</code> setting for an E2E transformer
[constr_3266]	<code>TransformationTechnology.hasInternalState</code> setting for a SOME/IP Transformer

[constr_3267]	PduTriggerings in Service Discovery SocketConnectionBundles
[constr_3268]	Service Discovery SocketConnectionBundle serverPort reference to a TpPort
[constr_3269]	Service Discovery SocketConnection clientPort reference to a TpPort
[constr_3270]	Service Discovery SocketConnection clientPort reference to an IP Address
[constr_3271]	clientIpAddrFromConnectionRequest and clientPortFromConnectionRequest settings for SD SocketConnections
[constr_3272]	SocketConnectionIpduIdentifier.headerId setting for SD SocketConnectionBundles
[constr_3273]	Service Discovery multicast SocketConnectionBundle's serverPort reference to an IP Address
[constr_3274]	Service Discovery unicast SocketConnectionBundle's serverPort reference to an IP Address
[constr_3275]	PduTriggering containment in different PdurIPduGroups of the same EcuInstance is not allowed
[constr_3276]	Prohibition of usage of allowedIPv6ExtHeaders in IPv4 SocketConnections
[constr_3277]	Restriction of usage of IPv6ExtHeaderFilterLists in IPv6 SocketConnections
[constr_3278]	Usage of SOMEIPTransformationProps.sizeOfArrayLengthField
[constr_3279]	Usage of SOMEIPTransformationProps.sizeOfStructLengthField
[constr_3280]	Usage of SOMEIPTransformationProps.sizeOfUnionLengthField
[constr_3281]	Usage of SOMEIPTransformationProps.alignment
[constr_3282]	SOME/IP Transformation settings for static size arrays in the context of an ISignal
[constr_3283]	SOME/IP Transformation settings for structures in the context of an ISignal
[constr_3284]	SOME/IP Transformation settings for unions in the context of an ISignal
[constr_3285]	Alignment of variable data length data elements in the context of an ISignal
[constr_3286]	ISignal.length shall be consistent to transformer configuration
[constr_3297]	Prohibition of usage of allowedTcpOptions in Udp SocketConnections
[constr_3298]	Ipv6Configuration.ipv6Address range in case of enableAnycast
[constr_3299]	SocketConnectionBundle.pathMtuDiscoveryEnabled setting dependency
[constr_3311]	Usage of SocketConnectionBundle.flowLabel
[constr_3312]	Consistency of vlanPriority and EthernetCommunicationConnector
[constr_3313]	E2E transformer configuration
[constr_3314]	BufferProperties.bufferComputation is mandatory
[constr_3315]	The value of V0 in BufferProperties.bufferComputation setting for a COM Based transformer
[constr_3316]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07
[constr_3317]	Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the ImplementationDataTypes
[constr_3318]	Allowed use of ISignal.networkRepresentationProps
[constr_3319]	Existence of DataPrototypeTransformationProps.networkRepresentationProps
[constr_3322]	Consistent setting of SocketConnectionIpduIdentifier.pduCollectionSemantics in the context of one SocketConnectionBundle
[constr_3323]	Relation between NmCluster.nmPncParticipation and PncMapping.pncGroup
[constr_3324]	Category of SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps
[constr_3325]	SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps attribute values for predefined categories
[constr_3326]	Allowed values for EndToEndTransformationISignalProps.dataIdMode in PROFILE_11

[constr_3327]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_22
[constr_3328]	<code>SomeIpTpConnection.transportPdu</code> reference restriction
[constr_3329]	<code>SomeIpTpConnection.tpSdu</code> reference restriction
[constr_3330]	Same <code>transportPdu</code> shall not be used in different <code>SomeIpTpConnections</code>
[constr_3331]	Standardized values for the attribute <code>category</code> of meta-class <code>EthernetCommunicationConnector</code>
[constr_3332]	Standardized values for the attribute <code>category</code> of meta-class <code>EthernetCommunicationController</code>
[constr_3333]	Standardized values for the attribute <code>category</code> of meta-class <code>EthernetPhysicalChannel</code>
[constr_3334]	Allowed references between <code>EthernetPhysicalChannel</code> and <code>EthernetCommunicationConnector</code>
[constr_3335]	Allowed references between <code>EthernetCommunicationConnector</code> and <code>EthernetCommunicationController</code>
[constr_3336]	<code>EthernetPhysicalChannel.soAdConfig</code> in case of WIRELESS <code>EthernetPhysicalChannel</code>
[constr_3337]	<code>IPduPort.useAuthDataFreshness</code> is configurable on the receiver side
[constr_3338]	<code>IPduPort.useAuthDataFreshness</code> validness
[constr_3339]	Relation between <code>authDataFreshnessStartPosition</code> , <code>authDataFreshnessLength</code> and <code>useAuthDataFreshness</code>
[constr_3521]	<code>defaultVlan</code> and <code>vlanMembership</code>
[constr_3522]	<code>vlanModifier</code> and <code>vlanMembership</code>
[constr_3523]	<code>CouplingPort</code> and <code>PncMapping</code> in the scope of an <code>EthernetPhysicalChannel</code>
[constr_3524]	Definition of <code>couplingPortRole</code> on <code>CouplingPort</code> for managed <code>CouplingElement</code>
[constr_3525]	Connection of <code>CouplingPort</code> with <code>couplingPortRole</code> set to <code>upLinkPort</code>

Table D.33: Added Constraints in 4.3.0

D.9.5 Changed Constraints in 4.3.0

Id	Heading
[constr_1372]	Consistency of attribute <code>globalTimePduTriggering</code>
[constr_3042]	<code>pncVectorLength</code> range
[constr_3049]	Role of <code>SystemSignal</code> in inter-ECU client server communication with clients located on different ECUs in case of networks other than Ethernet
[constr_3069]	Allowed <code>CanNmCluster.nmNidPosition</code> values
[constr_3070]	Allowed <code>CanNmCluster.nmCbvPosition</code> values
[constr_3078]	Allowed <code>UdpNmCluster.nmNidPosition</code> values
[constr_3079]	Allowed <code>UdpNmCluster.nmCbvPosition</code> values
[constr_3082]	Value of category in <code>GeneralPurposeIPdu</code>
[constr_3113]	<code>AbstractEthernetFrame</code> shall not have a <code>PduToFrameMapping</code>
[constr_3121]	The length of transformer chains is limited to 255 transformers
[constr_3128]	SOME/IP transformer configuration
[constr_3136]	Allowed payload of <code>SecuredIPdus</code>
[constr_3149]	<code>TransformationTechnology.needsOriginalData</code> settings for E2E Transformer
[constr_3150]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_01 and PROFILE_11 in case it is 0
[constr_3151]	<code>BufferProperties.headerLength</code> settings for an E2E transformer used in combination with a SOME/IP transformer

[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3154]	<code>BufferProperties.bufferComputation</code> setting for an E2E transformer when used together with a Com-based transformer
[constr_3156]	Allowed values for <code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_01 and PROFILE_11
[constr_3157]	Allowed values for <code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_01 and PROFILE_11 in case <code>dataIdMode</code> is set to <code>lower12Bit</code>
[constr_3158]	Allowed values for <code>EndToEndTransformationDescription.maxDeltaCounter</code> in PROFILE_01 and PROFILE_11
[constr_3160]	<code>EndToEndTransformationISignalProps.dataId</code> in PROFILE_02 and PROFILE_22
[constr_3161]	<code>EndToEndTransformationISignalProps.dataLength</code> in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22
[constr_3162]	<code>EndToEndTransformationISignalProps.minDataLength</code> and <code>EndToEndTransformationISignalProps.maxDataLength</code> in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22
[constr_3163]	<code>EndToEndTransformationISignalProps.minDataLength</code> and <code>EndToEndTransformationISignalProps.maxDataLength</code> in PROFILE_04, PROFILE_06, PROFILE_07
[constr_3164]	<code>EndToEndTransformationISignalProps.dataLength</code> in PROFILE_04, PROFILE_06, PROFILE_07
[constr_3165]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_01, PROFILE_11
[constr_3167]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[constr_3169]	Attribute multiplicities and values in PROFILE_02 and PROFILE_22
[constr_3171]	Value of <code>EndToEndTransformationISignalProps.dataId</code> shall be unique in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[constr_3174]	<code>EndToEndTransformationDescription</code> settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22
[constr_3184]	Only one <code>EndToEndTransformationISignalProps.dataId</code> element in PROFILE_01 and PROFILE_11
[constr_3185]	Multiplicity of <code>EndToEndTransformationDescription.dataIdMode</code> in PROFILE_01 and PROFILE_11
[constr_3186]	Multiplicity of <code>EndToEndTransformationDescription.dataIdMode</code> in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3187]	Multiplicity of <code>EndToEndTransformationDescription.counterOffset</code> in PROFILE_01 and PROFILE_11
[constr_3188]	Multiplicity of <code>EndToEndTransformationDescription.counterOffset</code> in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3189]	Multiplicity of <code>EndToEndTransformationDescription.crcOffset</code> in PROFILE_01 and PROFILE_11
[constr_3190]	Multiplicity of <code>EndToEndTransformationDescription.crcOffset</code> in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3191]	Multiplicity of <code>EndToEndTransformationDescription.dataIdNibbleOffset</code> in PROFILE_01, PROFILE_11 and <code>dataIdMode</code> equal to <code>lower12Bit</code>
[constr_3192]	Multiplicity of <code>EndToEndTransformationDescription.dataIdNibbleOffset</code> in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 or <code>dataIdMode</code> different from <code>lower12Bit</code>
[constr_3193]	Multiplicity of <code>EndToEndTransformationDescription.offset</code> in PROFILE_01 and PROFILE_11
[constr_3194]	Multiplicity of <code>EndToEndTransformationDescription.offset</code> in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22

[constr_3195]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02 and PROFILE_22
[constr_3514]	No two ISignalToIPduMappings shall reference the identical ISignal

Table D.34: Changed Constraints in 4.3.0

D.9.6 Deleted Constraints in 4.3.0

Id	Heading
[constr_3032]	Combinations of SwcToEcuMapping targets
[constr_3061]	CompuMethod specification in networkRepresentationProps
[constr_3207]	Assignment of SocketConnectionIpdulIdentifiers used for ClientServer Communication to SocketConnections

Table D.35: Deleted Constraints in 4.3.0

D.10 Constraint and Specification Item History of this document according to AUTOSAR R4.3.1

D.10.1 Added Traceables in 4.3.1

Number	Heading
[TPS_SYST_02162]	Routing of ISignals of ISignalGroups
[TPS_SYST_02163]	Applicability of syncLossTimeout
[TPS_SYST_02164]	LdCom: No ISignalPort.firstTimeout reception timeout defined
[TPS_SYST_02165]	Derivation of CanNmPnFilterMaskByte
[TPS_SYST_02166]	Derivation of UdpNmPnFilterMaskByte
[TPS_SYST_02167]	Derivation of FrNmPnFilterMaskByte
[TPS_SYST_02168]	MetaData support required if CanFrameTriggering.txMask is used
[TPS_SYST_02169]	MetaData support may be required if CanFrameTriggering.rxMask is used
[TPS_SYST_02170]	category of the GeneralPurposeConnection
[TPS_SYST_02171]	Secured Area in payload Pdu
[TPS_SYST_02172]	Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to false
[TPS_SYST_02173]	Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to true
[TPS_SYST_02174]	Initial Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02175]	Repetition Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02176]	Main Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02177]	TTL for Offer Service Entries





Number	Heading
[TPS_SYST_02178]	Servers RequestResponseDelay for received FindService entries
[TPS_SYST_02179]	Server Capability Records
[TPS_SYST_02180]	Usage of EventHandler.multicastThreshold
[TPS_SYST_02181]	TTL for SubscribeEventGroupAck Entries
[TPS_SYST_02182]	Servers RequestResponseDelay for received SubscribeEventGroup entries
[TPS_SYST_02183]	Initial Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02184]	Repetition Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02185]	TTL for Find Service Entries
[TPS_SYST_02186]	Client Capability Records
[TPS_SYST_02187]	SdClientConfig.ttl for SubscribeEventGroup Entries
[TPS_SYST_02188]	Clients RequestResponseDelay for received ServiceOffer entries
[TPS_SYST_02189]	Setting of useSecuredPduHeader attribute

Table D.36: Added Traceables in 4.3.1

D.10.2 Changed Traceables in 4.3.1

Number	Heading
[TPS_SYST_01120]	Precedence of ISignalMappings
[TPS_SYST_02098]	Header id and header type of a contained IPdu
[TPS_SYST_02100]	Relation between ContainerIPdu and contained IPdus on receiver side
[TPS_SYST_02112]	Usage of EventHandler.applicationEndpoint reference

Table D.37: Changed Traceables in 4.3.1

D.10.3 Deleted Traceables in 4.3.1

Number	Heading
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus
[TPS_SYST_02160]	EthernetPhysicalChannels with different category values are not allowed within an EthernetCluster

Table D.38: Deleted Traceables in 4.3.1

D.10.4 Added Constraints in 4.3.1

Number	Heading
[constr_3364]	<code>headerLength</code> shall be a multiple of 8
[constr_3365]	<code>EthernetPhysicalChannels</code> with different <code>category</code> values are not allowed within an <code>EthernetCluster</code>
[constr_3373]	Limitation on the number of <code>PhysicalChannels</code> that are referencing a <code>CommunicationConnector</code>
[constr_3378]	Maximal one <code>AliasNameAssignment</code> allowed per <code>FlatInstanceDescriptor</code>
[constr_3379]	Multiple <code>SocketAddress</code> entries with the same IP Address, Protocol and Port in the context of a given <code>EcuInstance</code>
[constr_3383]	Standardized values for the attribute <code>category</code> of meta-class <code>GeneralPurposeConnection</code>
[constr_3384]	<code>PduTriggerings</code> referenced by <code>GeneralPurposeConnection</code> shall be defined on the same <code>PhysicalChannel</code>
[constr_3385]	<code>XcpChannel</code> is allowed to reference exactly two <code>PduTriggerings</code>
[constr_3386]	<code>XcpChannel</code> is only allowed to reference <code>PduTriggerings</code> of <code>GeneralPurposeIPdus</code> with category XCP
[constr_3399]	Existence of <code>securedAreaOffset</code> and <code>securedAreaLength</code>
[constr_3400]	Usage of <code>SdClientConfig</code> attributes in <code>ConsumedServiceInstance</code> and <code>ConsumedEventGroup</code>
[constr_3401]	Usage of <code>SdServerConfig</code> attributes in <code>ProvidedServiceInstance</code> and <code>EventHandler</code>
[constr_3402]	Mandatory <code>offset</code> if <code>noHeader</code> is used
[constr_3403]	Usage of <code>ContainerIPdu.rxAcceptContainedIPdu</code> if <code>noHeader</code> is used
[constr_3404]	Usage of <code>ContainedIPduProps.updateIndicationBitPosition</code>
[constr_3405]	Dynamic Length <code>IPdu</code> inside of a static configured <code>ContainerIPdu</code>
[constr_3406]	All signals before <code>authDataFreshnessStartPosition</code> shall have a static length
[constr_3407]	Freshness Value in Authentic <code>IPdu</code> is not allowed to be used in case of <code>ContainerIPdu</code> with a dynamic layout

Table D.39: Added Constraints in 4.3.1

D.10.5 Changed Constraints in 4.3.1

Number	Heading
[constr_2025]	Uniqueness of <code>symbol</code> attributes
[constr_3052]	Complete <code>ISignalMapping</code> of <code>ISignalGroup</code> signals
[constr_3053]	Complete <code>ISignalMapping</code> of target <code>ISignalGroup</code>
[constr_3136]	Allowed payload of <code>SecuredIPdus</code>

Table D.40: Changed Constraints in 4.3.1

D.10.6 Deleted Constraints in 4.3.1

Number	Heading
[constr_3139]	Usage of IPduPort.rxSecurityVerification

Table D.41: Deleted Constraints in 4.3.1

D.11 Constraint and Specification Item History of this document according to AUTOSAR R4.4.0

D.11.1 Added Traceables in 4.4.0

Number	Heading
[TPS_SYST_02190]	J1939TpConnection.transmitter reference in case of broadcast connection
[TPS_SYST_02191]	J1939TpConnection.transmitter reference in case that the source is an unknown node
[TPS_SYST_02192]	J1939TpConnection.receiver reference in case that the destination is an unknown node
[TPS_SYST_02193]	J1939TpConnection.receiver reference in case that the destination is connected to a configured J1939NmNode
[TPS_SYST_02194]	Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[TPS_SYST_02195]	Applicable use cases for DataPrototypeInSystemRef
[TPS_SYST_02196]	PduTriggering is referenced by several ContainerIPdus
[TPS_SYST_02197]	DRAFT
[TPS_SYST_02198]	Applicable transferProperty for ISignal
[TPS_SYST_02199]	Applicable transferProperty for ISignalGroup and no group signal has transferProperty defined
[TPS_SYST_02200]	Applicable transferProperty for ISignalGroup and group signals have transferProperty defined
[TPS_SYST_02201]	Existence of CanFrameTriggering.identifier
[TPS_SYST_02202]	Modeling of bus mirroring
[TPS_SYST_02203]	BusMirroring to CAN destination channel
[TPS_SYST_02204]	BusMirroring to FlexRay destination channel
[TPS_SYST_02205]	BusMirroring to Ethernet destination channel
[TPS_SYST_02206]	BusMirroring to UserDefined destination channel
[TPS_SYST_05016]	Assignment of TLV data ids
[TPS_SYST_05017]	Definition of the applicable wire type attribute SOMEIPTransformation-ISignalProps.isDynamicLengthFieldSize shall be used to define the applicable wire type
[TPS_SYST_05018]	Semantics of meta-class LinSlave



△

Number	Heading
[TPS_SYST_05019]	Semantics of <code>LinErrorResponse.responseError</code>
[TPS_SYST_05020]	Semantics of <code>CryptoServiceMapping</code>
[TPS_SYST_05021]	Semantics of <code>SecOcCryptoServiceMapping</code>
[TPS_SYST_05022]	Semantics of <code>PduTriggering.secOcCryptoMapping</code>
[TPS_SYST_05023]	Semantics of <code>CryptoServicePrimitive</code>
[TPS_SYST_05024]	Semantics of <code>CryptoServiceKey</code>
[TPS_SYST_05025]	Standardized values of <code>CryptoServicePrimitive.algorithmFamily</code> and <code>CryptoServiceKey.algorithmFamily</code>
[TPS_SYST_05026]	Relation of <code>CryptoServicePrimitive.algorithmFamily</code> to <code>CryptoServiceKey.algorithmFamily</code>
[TPS_SYST_05027]	Standardized values of <code>CryptoServicePrimitive.algorithmMode</code>
[TPS_SYST_05028]	Semantics of <code>CryptoServiceKey.keyStorageType</code>
[TPS_SYST_05029]	Semantics of meta-class <code>TlsCryptoServiceMapping</code>
[TPS_SYST_05030]	Semantics of <code>TlsCryptoCipherSuite</code>
[TPS_SYST_05031]	Existence of <code>TlsCryptoCipherSuite.keyExchange</code> vs. <code>TlsCryptoServiceMapping.keyExchange</code>
[TPS_SYST_05032]	Semantics of <code>CryptoServiceCertificate</code>
[TPS_SYST_05033]	Existence of <code>TlsCryptoCipherSuite.certificate</code> in the <i>client</i> role

Table D.42: Added Traceables in 4.4.0

D.11.2 Changed Traceables in 4.4.0

Number	Heading
[TPS_SYST_01100]	TP routing without using transport protocol modules (low-level routing)
[TPS_SYST_01101]	TP routing using transport protocol modules
[TPS_SYST_01114]	Frame fan-out support
[TPS_SYST_02064]	Reception acceptance of contained <code>IPdus</code>
[TPS_SYST_02068]	Transformer header field representation in an <code>ISignalGroup</code>
[TPS_SYST_02099]	Relation between <code>ContainerIPdu</code> and contained <code>PduTriggerings</code> on sender side
[TPS_SYST_02113]	Usage of <code>ConsumedEventGroup.applicationEndpoint</code> reference
[TPS_SYST_02131]	Assignment of <code>SOMEIPTransformationProps</code> to a root <code>AutosarDataPrototype</code> typed by an <code>ImplementationDataType</code>

Table D.43: Changed Traceables in 4.4.0

D.11.3 Deleted Traceables in 4.4.0

Number	Heading
[TPS_SYST_01046]	ShortNames of LinSlaveConfig and LinSlave
[TPS_SYST_01049]	Handling of IPdus with dynamic signals
[TPS_SYST_02112]	Usage of EventHandler .applicationEndpoint reference

Table D.44: Deleted Traceables in 4.4.0

D.11.4 Added Constraints in 4.4.0

Number	Heading
[constr_1641]	Consistent assignment of TLV data ids to ApplicationRecordDataType
[constr_1642]	Consistent assignment of TLV data ids to ImplementationDataType or ImplementationDataTypeElement
[constr_1643]	Completeness of the existence of a set of TlvDataIdDefinition.tlvArguments
[constr_1644]	Completeness of the existence of a set of TlvDataIdDefinition.tlvRecordElements
[constr_1645]	Completeness of the existence of a set of TlvDataIdDefinition.tlvImplementationDataTypeElements
[constr_1646]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ArgumentDataPrototype
[constr_1647]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ApplicationRecordElement
[constr_1648]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ImplementationDataTypeElement
[constr_1649]	TlvDataIdDefinition referencing ArgumentDataPrototype
[constr_1650]	TlvDataIdDefinition referencing ApplicationRecordElement
[constr_1651]	TlvDataIdDefinition referencing ImplementationDataTypeElement
[constr_1652]	Definition of static length fields sizes in case of TLV usage
[constr_1653]	Identical values for length fields sizes in case of TLV usage
[constr_1654]	No definition of length field sizes on DataPrototype level in case of TLV usage
[constr_1655]	The mutual existence of LinMasters in the LinSlave EcuExtract
[constr_1656]	No application-level write access to LinErrorResponse.responseError on Lin slave
[constr_1657]	Existence of LinPhysicalChannel.scheduleTable
[constr_1669]	Existence of PduTriggering.secOcCryptoMapping
[constr_1670]	Prohibition of usage of tlsCryptoMapping in case of UDP socket connections
[constr_1671]	Supported values of TlsCryptoServiceMapping.category
[constr_1672]	Existence of TlsCryptoCipherSuite.certificate in the <i>server</i> role
[constr_3435]	Applicability of CouplingPort.macMulticastAddress





Number	Heading
[constr_3436]	Value range of <code>minimumTxContainerQueueSize</code> and <code>minimumRxContainerQueueSize</code>
[constr_3437]	<code>invalidValue</code> defined in the context of <code>ISignal</code>
[constr_3438]	<code>timeoutSubstitutionValue</code> defined in the context of <code>ISignal</code>
[constr_3448]	Restriction for usage of <code>Pdu.hasDynamicLength</code>
[constr_3453]	Uniqueness of header <code>Id</code> in case of <code>acceptAll ContainerIPdu</code>
[constr_3454]	Unique <code>headerIdLongHeader</code> for <code>acceptConfigured</code>
[constr_3455]	Unique <code>headerIdShortHeader</code> for <code>acceptConfigured</code>
[constr_3456]	Existence of <code>ProvidedServiceInstance.loadBalancingPriority</code> and <code>ProvidedServiceInstance.loadBalancingWeight</code>
[constr_3457]	Uniqueness of <code>ConsumedEventGroup.eventGroupIdentifier</code> in the scope of a <code>ConsumedServiceInstance</code>
[constr_3458]	DRAFT
[constr_3459]	Applicable <code>transferProperty</code> for group signal
[constr_3460]	Full definition of <code>transferProperty</code> for group signal
[constr_3461]	<code>TransferProperty</code> for group signals if <code>ISignalGroup</code> has <code>transferProperty=pending</code>
[constr_3464]	Allowed Pdu type on <code>BusMirrorChannelMapping.targetChannel</code>
[constr_3465]	Identical <code>BusMirrorChannel.busMirrorNetworkId</code> for <code>BusMirrorChannels</code> referencing the same <code>PhysicalChannel</code>
[constr_3466]	Unique <code>BusMirrorChannel.busMirrorNetworkIds</code> for each specialization of <code>PhysicalChannel</code>
[constr_3467]	<code>CanPhysicalChannel</code> as destination channel of <code>BusMirrorChannelMapping-Can</code>
[constr_3468]	<code>BusMirrorChannelMappingCan.targetPduTriggering</code> restriction
[constr_3469]	<code>CanFrameTriggering.txMask</code> setting for the destination frame
[constr_3470]	<code>PaddingValue</code> used to transmit the Pdu on a Can-Fd destination bus
[constr_3471]	<code>FlexrayPhysicalChannel</code> as destination channel of <code>BusMirrorChannelMappingFlexray</code>
[constr_3472]	Number of <code>BusMirrorChannels</code> derived for one <code>FlexrayCluster</code>
[constr_3473]	<code>BusMirrorChannelMappingFlexray.targetPduTriggering</code> restriction
[constr_3474]	<code>EthernetPhysicalChannel</code> as destination channel of <code>BusMirrorChannelMappingIp</code>
[constr_3475]	<code>BusMirrorChannelMappingIp.targetPduTriggering</code> restriction
[constr_3476]	<code>UserDefinedPhysicalChannel</code> as destination channel of <code>BusMirrorChannelMappingUserDefined</code>
[constr_3477]	<code>BusMirrorChannelMappingUserDefined.targetPduTriggering</code> restriction
[constr_3479]	<code>PhysicalChannel</code> is not allowed to be a <code>managedPhysicalChannel</code> and a managing <code>PhysicalChannel</code>
[constr_3480]	<code>PhysicalChannel</code> shall be referenced in the role <code>managedPhysicalChannel</code> only once





Number	Heading
[constr_3481]	<code>UdpNmCluster</code> is not allowed to reference a <code>managedPhysicalChannel</code> in the role <code>vlan</code>
[constr_3482]	<code>NmCluster</code> is not allowed to reference a <code>CommunicationCluster</code> that aggregates a <code>managedPhysicalChannel</code>
[constr_3483]	The same <code>PhysicalChannel</code> is not allowed to be the source and the target of <code>managedPhysicalChannel</code> references
[constr_3484]	<code>PncMapping</code> that refers a <code>managedPhysicalChannel</code> shall also refer the managing <code>PhysicalChannel</code>
[constr_3488]	Value range of <code>ContainedIPduProps.priority</code>
[constr_3489]	<code>ContainedIPduProps.priority</code> is only applicable if a <code>ContainerIPdu</code> header is used
[constr_3490]	<code>ContainedIPduProps.priority</code> is only applicable if <code>collectionSemantics</code> is set to <code>lastIsBest</code>
[constr_3533]	<code>EndToEndTransformationISignalProps.dataLength</code> shall be a multiple of 8
[constr_3534]	<code>EthernetPhysicalChannel</code> shall only be referenced by one <code>VlanMembership</code>
[constr_3535]	<code>EthernetCommunicationController</code> shall aggregate at most one <code>CouplingPort</code>

Table D.45: Added Constraints in 4.4.0

D.11.5 Changed Constraints in 4.4.0

Number	Heading
[constr_3067]	<code>initValue</code> defined in the context of <code>ISignal</code>
[constr_3165]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in <code>PROFILE_01</code> , <code>PROFILE_11</code>
[constr_3219]	The mutual existence of <code>LinSlaves</code> in the <code>LinMaster</code> <code>EcuExtract</code>
[constr_3323]	Relation between <code>NmCluster.nmPncParticipation</code> and <code>PncMapping.pncGroup</code>
[constr_3327]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in <code>PROFILE_22</code>

Table D.46: Changed Constraints in 4.4.0

D.11.6 Deleted Constraints in 4.4.0

Number	Heading
[constr_3024]	Usage of <code>triggeredWithoutRepetition</code> and <code>triggeredOnChangeWithoutRepetition</code> is not allowed for signal groups and group signals.
[constr_3034]	Values of <code>LinSlaveConfig</code> and <code>LinSlave</code> attributes
[constr_3043]	<code>pncVector</code> configuration in AUTOSAR Com





Number	Heading
[constr_3150]	Effect of <code>EndToEndTransformationDescription.upperHeaderBitsToShift</code> value in PROFILE_01 and PROFILE_11 in case it is 0
[constr_3171]	Value of <code>EndToEndTransformationISignalProps.dataId</code> shall be unique in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07

Table D.47: Deleted Constraints in 4.4.0

D.12 Constraint and Specification Item History of this document according to AUTOSAR R19-11

D.12.1 Added Traceables in 19-11

Number	Heading
[TPS_SYST_02207]	Routing on the fly
[TPS_SYST_02208]	<code>ISignalPort.handleInvalid</code> defines the reception invalidation behavior
[TPS_SYST_02209]	Not defined <code>ISignalPort.handleInvalid</code> behavior
[TPS_SYST_02210]	Data invalidation in case the <code>dataTypePolicy</code> is set to <code>override</code> or <code>legacy</code>
[TPS_SYST_02211]	Reference from <code>SOMEIPTransformationISignalProps</code> to <code>TlvDataId-DefinitionSet</code>
[TPS_SYST_02212]	Assignment of <code>SOMEIPTransformationProps</code> to a root <code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ApplicationDataType</code>
[TPS_SYST_02213]	Assignment of <code>SOMEIPTransformationProps</code> to a subElement of a root <code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ApplicationDataType</code>
[TPS_SYST_02214]	Assignment of <code>SOMEIPTransformationProps</code> to a root <code>AutosarDataPrototype</code> in a <code>ClientServerInterface</code> typed by an <code>ImplementationDataType</code>
[TPS_SYST_02215]	Usage of <code>portNumber</code> with value 0
[TPS_SYST_02216]	Configuration of <code>capabilityRecords</code>
[TPS_SYST_02217]	SOME/IP Service offer
[TPS_SYST_02218]	<code>ProvidedServiceInstance</code> deployment
[TPS_SYST_02219]	Static configuration between <code>ProvidedServiceInstance</code> and <code>ConsumedServiceInstance</code>
[TPS_SYST_02220]	Maximal number of clients that may connect to the local server address
[TPS_SYST_02221]	<code>ProvidedServiceInstance.localUnicastAddress</code> reference target
[TPS_SYST_02222]	Usage of <code>headerId</code>
[TPS_SYST_02223]	Activation/Deactivation of <code>PduActivationRoutingGroups</code>
[TPS_SYST_02224]	Methods provided by a <code>ProvidedServiceInstance</code>





Number	Heading
[TPS_SYST_02225]	Service methods provided over UDP
[TPS_SYST_02226]	Service methods provided over TCP
[TPS_SYST_02227]	Publishing of a SOME/IP Event group
[TPS_SYST_02228]	Transmission of events over UDP/TCP Port
[TPS_SYST_02229]	Event groups provided by a ProvidedServiceInstance
[TPS_SYST_02230]	PduActivationRoutingGroups for event groups
[TPS_SYST_02231]	PduActivationRoutingGroups for methods
[TPS_SYST_02232]	Events provided over UDP
[TPS_SYST_02233]	Events provided over TCP
[TPS_SYST_02234]	SOME/IP Service search
[TPS_SYST_02235]	ConsumedServiceInstance deployment
[TPS_SYST_02236]	Static configuration between ConsumedServiceInstance and ProvidedServiceInstance
[TPS_SYST_02237]	Maximal number of servers that may connect to the local client address
[TPS_SYST_02238]	ConsumedServiceInstance.localUnicastAddress reference target
[TPS_SYST_02239]	Methods consumed by a ConsumedServiceInstance
[TPS_SYST_02240]	Service methods consumed over UDP
[TPS_SYST_02241]	Service methods consumed over TCP
[TPS_SYST_02242]	Subscription to a SOME/IP Event group
[TPS_SYST_02243]	Reception of events over UDP/TCP Port in case of Service Discovery
[TPS_SYST_02244]	ConsumedServiceInstance without a defined localUnicastAddress
[TPS_SYST_02245]	Event groups consumed by a ConsumedServiceInstance
[TPS_SYST_02246]	PduActivationRoutingGroups for ConsumedEventGroups
[TPS_SYST_02247]	Events consumed over UDP
[TPS_SYST_02248]	Events consumed over TCP
[TPS_SYST_02249]	Service Discovery Message sending behavior on ProvidedServiceInstance
[TPS_SYST_02250]	Service Discovery Message sending behavior on ConsumedServiceInstance
[TPS_SYST_02251]	Non-SOME/IP data exchange between two communication endpoints
[TPS_SYST_02252]	Description of a TCP Client
[TPS_SYST_02253]	Description of a TCP Server
[TPS_SYST_02254]	Pdus transported over the StaticSocketConnection
[TPS_SYST_02255]	Frame.frameLength usage for FlexrayFrames and CanFrames
[TPS_SYST_02256]	Allowed CanFrame.frameLength settings
[TPS_SYST_02257]	Standardized values of LinCommunicationController.protocolVersion and LinSlaveConfig.protocolVersion
[TPS_SYST_02258]	Shared random timer for ProvidedServiceInstance service discovery
[TPS_SYST_02259]	Shared random timer for ProvidedServiceInstance service discovery





Number	Heading
[TPS_SYST_02260]	Individual random timer for <code>ProvidedServiceInstance</code> service discovery
[TPS_SYST_02261]	Shared random timer for <code>ConsumedServiceInstance</code> service discovery
[TPS_SYST_02262]	Shared random timer for <code>ConsumedServiceInstance</code> service discovery
[TPS_SYST_02263]	Individual random timer for <code>ConsumedServiceInstance</code> service discovery
[TPS_SYST_02264]	Usage of <code>DltLogChannel</code>
[TPS_SYST_02265]	Configuration of IPsec
[TPS_SYST_02266]	Definition of <code>IPSecRules</code>
[TPS_SYST_02267]	IPsec policy
[TPS_SYST_02268]	IPsec mode
[TPS_SYST_02269]	IPsec AH and ESP protocol configuration
[TPS_SYST_02270]	Definition of general IPsec configuration settings
[TPS_SYST_02271]	IPsec AH and ESP CipherSuites
[TPS_SYST_02272]	IPsec Internet Key Exchange protocol configuration
[TPS_SYST_02273]	Protection of <code>ProvidedServiceInstance</code> by IPsec
[TPS_SYST_02274]	Protection of <code>ConsumedServiceInstance</code> by IPsec
[TPS_SYST_03022]	Provided service instance with <code>translationStart</code> requires <code>SdServerServiceAutoAvailable</code>
[TPS_SYST_03023]	Required service instance with <code>translationStart</code> requires <code>SdClientServiceAutoRequire</code>
[TPS_SYST_03024]	Required service instance with <code>translationStart</code> requires <code>SdConsumedEventGroupAutoRequire</code>
[TPS_SYST_03025]	Control of service instance in case of <code>partialNetwork</code>
[TPS_SYST_03026]	Monitoring of the <i>partial networks</i> status in case of <code>partialNetwork</code> for provided service instance
[TPS_SYST_03027]	Monitoring of the <i>partial networks</i> status in case of <code>partialNetwork</code> for required service instance
[TPS_SYST_03028]	Initial <i>service find</i> for a provided service instance with <code>serviceControl</code>
[TPS_SYST_03029]	Initial <i>offer</i> for a provided service instance with <code>serviceControl</code>
[TPS_SYST_03030]	<i>Stop offer</i> for a provided service instance with <code>serviceControl</code>
[TPS_SYST_03031]	Sufficient ASIL level of translation software component
[TPS_SYST_03032]	Data transmission in case of <code>E_OK</code> safe signal reception
[TPS_SYST_03033]	No data transmission in case of reception timeout
[TPS_SYST_03034]	Handling safe signal reception
[TPS_SYST_03036]	<code>PortAPIOption</code> for <code>safeTranslation RPortPrototype</code>
[TPS_SYST_03037]	<code>PortAPIOption</code> for <code>safeTranslation PPortPrototype</code>
[TPS_SYST_03038]	Definition of transmission triggers for translations with different sources
[TPS_SYST_03039]	Full translation before transmission triggering
[TPS_SYST_03040]	Transmission trigger for translations with different sources





Number	Heading
[TPS_SYST_03041]	No transmission trigger for translations with different sources
[TPS_SYST_03042]	Periodic call in case of ReceiverComSpec.dataUpdatePeriod
[TPS_SYST_03043]	Periodic call in case of SenderComSpec.dataUpdatePeriod
[TPS_SYST_03044]	Handling of safe signal-service-translation in one software component
[TPS_SYST_03045]	Support for safe signal-service-translation and service-signal-translation
[TPS_SYST_03046]	Support for safe signal-service-translation with same or different E2E profiles
[TPS_SYST_03047]	1:n mapping for E2E protected data
[TPS_SYST_03048]	E2E protected target out of E2E protected sources
[TPS_SYST_03049]	No translation of not OK E2E protected composed data
[TPS_SYST_03050]	Usage of ConsumedServiceInstance.blacklistedVersion
[TPS_SYST_03051]	Data filter inside the signal-service-translation software component
[TPS_SYST_05034]	DataMapping of ImplementationDataType of category UNION, DATA_REFERENCE, or FUNCTION_REFERENCE

Table D.48: Added Traceables in 19-11

D.12.2 Changed Traceables in 19-11

Number	Heading
[TPS_SYST_01090]	valid NetworkEndpoint
[TPS_SYST_01143]	DataMapping on the sender side for elements of a composite data type
[TPS_SYST_02019]	LdCom: ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition for sent ISignals
[TPS_SYST_02020]	LdCom: No IPduTiming.minimumDelay defined
[TPS_SYST_02022]	LdCom: Only the transmissionModeTrueTiming defined
[TPS_SYST_02023]	LdCom: DataFilter "always" if TransmissionModeCondition defined
[TPS_SYST_02024]	LdCom: No ModeDrivenTransmissionModeCondition defined
[TPS_SYST_02119]	StaticSocketConnections for GeneralPurposePdus with category SD
[TPS_SYST_02129]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType
[TPS_SYST_02130]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType
[TPS_SYST_02131]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ImplementationDataType
[TPS_SYST_02132]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ImplementationDataType





Number	Heading
[TPS_SYST_02133]	<code>BufferProperties.bufferComputation</code> setting for a COM Based transformer
[TPS_SYST_02140]	<code>SocketAddress.udpChecksumHandling</code> default value
[TPS_SYST_02141]	Semantics of <code>udpChecksumHandling</code>
[TPS_SYST_02142]	Reception of invalid checksum
[TPS_SYST_02150]	Role of <code>SystemSignal</code> in inter-ECU client server communication over Ethernet with clients located on different ECUs
[TPS_SYST_02151]	MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs
[TPS_SYST_02174]	Initial Wait Phase configuration for a <code>ProvidedServiceInstance</code>
[TPS_SYST_02175]	Repetition Wait Phase configuration for a <code>ProvidedServiceInstance</code>
[TPS_SYST_02176]	Main Phase configuration for a <code>ProvidedServiceInstance</code>
[TPS_SYST_02177]	TTL for Offer Service Entries
[TPS_SYST_02178]	Servers <code>RequestResponseDelay</code> for received <code>FindService</code> entries
[TPS_SYST_02182]	Servers <code>RequestResponseDelay</code> for received <code>SubscribeEventGroup</code> entries
[TPS_SYST_02183]	Initial Wait Phase configuration for a <code>ConsumedServiceInstance</code>
[TPS_SYST_02184]	Repetition Wait Phase configuration for a <code>ConsumedServiceInstance</code>
[TPS_SYST_02185]	TTL for Find Service Entries
[TPS_SYST_02187]	<code>SomeipSdClientEventGroupTimingConfig.timeToLive</code> for <code>SubscribeEventGroup</code> Entries
[TPS_SYST_02188]	Clients <code>RequestResponseDelay</code> for received <code>ServiceOffer</code> entries
[TPS_SYST_02189]	Setting of <code>useSecuredPduHeader</code> attribute
[TPS_SYST_02195]	Applicable use cases for <code>DataPrototypeReference</code>
[TPS_SYST_05016]	Assignment of TLV data ids
[TPS_SYST_05033]	Existence of <code>TlsCryptoCipherSuite.certificate</code> and <code>TlsCryptoCipherSuite.pskIdentity</code> in the <i>client</i> role

Table D.49: Changed Traceables in 19-11

D.12.3 Deleted Traceables in 19-11

Number	Heading
[TPS_SYST_01092]	Transmission of multiple <code>Pdus</code> over the same <code>SocketConnection</code>
[TPS_SYST_01093]	Activation/Deactivation of <code>SoAdRoutingGroups</code>
[TPS_SYST_01151]	<code>DataMapping</code> reference to an <code>EventHandler</code>
[TPS_SYST_01152]	<code>DataMapping</code> reference to a <code>ConsumedEventGroup</code>
[TPS_SYST_02002]	<code>SoAdRoutingGroup</code> for Services with Methods
[TPS_SYST_02003]	<code>SoAdRoutingGroups</code> for Services with event groups





Number	Heading
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02007]	Usage of SocketConnection attributes in the unicast server view
[TPS_SYST_02008]	Usage of SocketConnection attributes in the unicast client view
[TPS_SYST_02009]	Usage of SocketConnection attributes in the multicast server view
[TPS_SYST_02010]	Usage of SocketConnection attributes in the multicast client view
[TPS_SYST_02096]	Sending of ANY finds for minor version
[TPS_SYST_02113]	Usage of ConsumedEventGroup.applicationEndpoint reference
[TPS_SYST_02120]	runtimeIpAddressConfiguration and runtimePortConfiguration settings for SD SocketConnections
[TPS_SYST_02161]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that Com is used
[TPS_SYST_02179]	Server Capability Records
[TPS_SYST_02181]	TTL for SubscribeEventGroupAck Entries
[TPS_SYST_02186]	Client Capability Records
[TPS_SYST_05005]	Relation of GlobalTimeDomain to CommunicationCluster

Table D.50: Deleted Traceables in 19-11

D.12.4 Added Constraints in 19-11

Number	Heading
[constr_3545]	Mandatory reference to a Pnc in case of partialNetwork
[constr_3546]	Mandatory reference to a ConsumedEventGroup in case of serviceControl
[constr_3547]	Mandatory reference to an EventHandler in case of serviceControl
[constr_3548]	EndToEnd profile for both ends of safeTranslation
[constr_3549]	Secure payload for both ends in case of secureTranslation
[constr_3559]	ConsumedServiceInstance.blacklistedVersion is restricted to the usage of minorVersion
[constr_3560]	minimumMinorVersion and ConsumedServiceInstance.minorVersion value
[constr_5029]	J1939NmCluster is not allowed to reference a TtcanCluster
[constr_5030]	Uniqueness of LinOrderedConfigurableFrame.index
[constr_5031]	Uniqueness of FramePid.index
[constr_5032]	Maximal one NmConfig per System is allowed to be defined
[constr_5049]	Ethernet switch packet to traffic class assignment restriction
[constr_5050]	VariableDataPrototype of COM Based Transformer
[constr_5051]	Existence of CanFrameTriggering.identifier in case of bus mirror target
[constr_5053]	Existence of ISignalPort.handleInvalid





Number	Heading
[constr_5054]	<code>externalReplacement</code> not applicable for <code>ISignalPort.handleInvalid</code>
[constr_5055]	<code>DataMapping</code> of elements of <code>PRPortPrototypes</code> is not supported
[constr_5058]	Value range for <code>CryptoServiceQueue.queueSize</code>
[constr_5060]	Mapping of a <code>SecuredIPdu</code> into a <code>LinFrame</code> is not allowed
[constr_5061]	<code>EthernetCommunicationConnectors</code> and referencing <code>SocketAddresses</code> shall be in the same VLAN
[constr_5062]	SOME/IP <code>ProvidedServiceInstances</code> of the same <code>serviceInterface</code> on one <code>EcuInstance</code>
[constr_5063]	<code>ProvidedServiceInstance.serviceIdentifier</code> is mandatory
[constr_5064]	<code>ProvidedServiceInstance.majorVersion</code> is mandatory
[constr_5065]	<code>ProvidedServiceInstance.minorVersion</code> is mandatory
[constr_5066]	<code>ProvidedServiceInstance.instanceIdentifier</code> is mandatory
[constr_5067]	<code>ProvidedServiceInstance</code> shall be unique in respect of <code>serviceIdentifier</code> , <code>instanceIdentifier</code> , <code>majorVersion</code>
[constr_5068]	<code>ProvidedServiceInstance.localUnicastAddress</code> shall be IP Unicast
[constr_5069]	<code>ProvidedServiceInstance.remoteUnicastAddress</code> shall be IP Unicast
[constr_5070]	<code>headerIds</code> of <code>ProvidedServiceInstances</code> shall be unique on a <code>SocketAddress</code> per communication direction
[constr_5071]	<code>EventHandler.eventMulticastAddress</code> reference target
[constr_5072]	<code>EventHandler</code> without defined <code>eventMulticastAddress</code>
[constr_5073]	<code>PduActivationRoutingGroup</code> with <code>eventGroupControlType</code> set to <code>activationUnicast</code> or <code>triggerUnicast</code> or <code>activationAndTriggerUnicast</code> that is aggregated by an <code>EventHandler</code>
[constr_5074]	<code>PduActivationRoutingGroup</code> with <code>eventGroupControlType</code> set to <code>activationMulticast</code> that is aggregated by an <code>EventHandler</code>
[constr_5075]	Allowed references of <code>SoConIPduIdentifiers</code> by <code>PduActivationRoutingGroup</code> with <code>eventGroupControlType</code> set to <code>activationMulticast</code> and allowed <code>SoConIPduIdentifier</code> references
[constr_5076]	<code>PduActivationRoutingGroup</code> with <code>iPduIdentifierTcp</code> reference that is aggregated by a <code>ProvidedServiceInstance</code>
[constr_5077]	<code>PduActivationRoutingGroup</code> with <code>iPduIdentifierUdp</code> reference that is aggregated by a <code>ProvidedServiceInstance</code>
[constr_5078]	<code>PduTriggerings</code> referenced by a <code>PduActivationRoutingGroup</code> shall be on the same VLAN as the referencing <code>PduActivationRoutingGroup</code>
[constr_5079]	Service communication is restricted to one VLAN
[constr_5080]	<code>ApplicationEndpoints</code> referenced by <code>EventHandlers</code> and by the aggregating <code>ProvidedServiceInstance</code> shall be in the same VLAN
[constr_5081]	<code>ConsumedServiceInstance.serviceIdentifier</code> is mandatory
[constr_5082]	<code>ConsumedServiceInstance.majorVersion</code> is mandatory
[constr_5083]	<code>ConsumedServiceInstance.minorVersion</code> is mandatory
[constr_5084]	<code>ConsumedServiceInstance.instanceIdentifier</code> is mandatory





Number	Heading
[constr_5085]	<code>ConsumedServiceInstance.localUnicastAddress</code> shall be IP Unicast
[constr_5086]	<code>ConsumedServiceInstance.remoteUnicastAddress</code> shall be IP Unicast
[constr_5087]	<code>PduActivationRoutingGroup</code> with <code>eventGroupControlType</code> set to <code>activationUnicast</code> or <code>triggerUnicast</code> or <code>activationAndTriggerUnicast</code> that is referenced by a <code>ConsumedEventGroup</code>
[constr_5088]	<code>PduActivationRoutingGroup</code> with <code>iPduIdentifierTcp</code> reference that is aggregated by a <code>ConsumedServiceInstance</code>
[constr_5089]	<code>PduActivationRoutingGroup</code> with <code>iPduIdentifierUdp</code> reference that is aggregated by a <code>ConsumedServiceInstance</code>
[constr_5090]	<code>ApplicationEndpoints</code> referenced by <code>ConsumedEventGroups</code> and by the aggregating <code>ConsumedServiceInstance</code> shall be in the same VLAN
[constr_5091]	Relevance of <code>tcpRole</code> attribute
[constr_5092]	Relevance of <code>tcpRole</code> attribute
[constr_5093]	<code>pncGatewayType</code> and <code>PhysicalChannel</code>
[constr_5094]	<code>pncGatewayType</code> and ECU
[constr_5095]	Relationship between the timing behavior of the <code>ConsumedEventGroup</code> retry and the timing behavior of an Offer message
[constr_5096]	<code>ConsumedEventGroup</code> with value <code>subscribeEventgroupRetryMax</code> set to 255
[constr_5097]	<code>DltLogChannel.txPduTriggering</code> and <code>DltLogChannel.rxPduTriggering</code> shall point to <code>GeneralPurposeIPdus</code> of category DLT
[constr_5098]	Usage of <code>DltArgument.networkRepresentation</code>
[constr_5099]	Standardized values of <code>DltMessage.messageTypeInfo</code>
[constr_5100]	Compatibility of two <code>MetaDataItemSets</code>
[constr_5101]	Consistent Definition of meta-data
[constr_5104]	Assignment of a <code>FlexrayFrame</code> where <code>allowDynamicLSduLength</code> is set to true
[constr_5105]	Mapping of <code>Pdu</code> with dynamic length in a <code>FlexrayFrame</code>

Table D.51: Added Constraints in 19-11

D.12.5 Changed Constraints in 19-11

Number	Heading
[constr_1207]	Existence of the attribute <code>DataMapping.communicationDirection</code> in the context of a <code>SenderReceiverInterface</code> or <code>TriggerInterface</code>
[constr_1441]	In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer
[constr_1643]	Completeness of the existence of a set of <code>TlvDataIdDefinition.tlvArguments</code>
[constr_1644]	Completeness of the existence of a set of <code>TlvDataIdDefinition.tlvRecordElements</code>



△

Number	Heading
[constr_1645]	Completeness of the existence of a set of <code>TlvDataIdDefinition.tlvImplementationDataElements</code>
[constr_1672]	Existence of <code>TlsCryptoCipherSuite.certificate</code> and <code>TlsCryptoCipherSuite.pskIdentity</code> in the <i>server</i> role
[constr_3012]	Overlapping of <code>Pdus</code> is prohibited
[constr_3013]	<code>FlexrayFrame</code> length shall not be exceeded
[constr_3014]	Overlapping of <code>updateIndicationBits</code> for <code>Pdus</code> is prohibited
[constr_3267]	<code>PduTriggerings</code> in Service Discovery <code>StaticSocketConnections</code>
[constr_3268]	Service Discovery <code>StaticSocketConnection</code> aggregation by an <code>ApplicationEndpoint</code>
[constr_3269]	Service Discovery <code>StaticSocketConnection</code> <code>remoteAddress</code> reference to a <code>TpPort</code>
[constr_3270]	Service Discovery <code>SocketConnection</code> <code>remoteAddress</code> reference to an IP Address
[constr_3272]	<code>SoConIPduIdentifier.headerId</code> setting for SD <code>StaticSocketConnections</code>
[constr_3273]	Service Discovery multicast <code>StaticSocketConnection</code> 's aggregation by an <code>ApplicationEndpoint</code>
[constr_3274]	Service Discovery unicast <code>StaticSocketConnection</code> 's aggregation by an <code>ApplicationEndpoint</code>
[constr_3276]	Prohibition of usage of <code>allowedIPv6ExtHeaders</code> in IPv4 <code>SocketAddress</code>
[constr_3277]	Restriction of usage of <code>IPv6ExtHeaderFilterLists</code> in IPv6 <code>SocketAddress</code>
[constr_3297]	Prohibition of usage of <code>allowedTcpOptions</code> in <code>Udp</code> <code>SocketAddress</code>
[constr_3299]	<code>SocketAddress.pathMtuDiscoveryEnabled</code> setting dependency
[constr_3311]	Usage of <code>SocketAddress.flowLabel</code>
[constr_3315]	The value of <code>V0</code> in <code>BufferProperties.bufferComputation</code> setting for a COM Based transformer
[constr_3322]	Consistent setting of <code>SoConIPduIdentifier.pduCollectionSemantics</code> in the context of one <code>SocketAddress</code>
[constr_3325]	<code>SecureCommunicationFreshnessProps</code> , <code>SecureCommunicationAuthenticationProps</code> and <code>CryptoServicePrimitive</code> attribute values for predefined categories
[constr_3379]	Multiple <code>SocketAddress</code> entries with the same IP Address, Protocol and Port in the context of a given <code>EcuInstance</code>
[constr_3506]	Mapping of composite data type to <code>SystemSignals</code> in <code>SystemSignalGroup</code>
[constr_3516]	limitation of <code>Frame.frameLength</code> for CAN L-PDUs

Table D.52: Changed Constraints in 19-11

D.12.6 Deleted Constraints in 19-11

Number	Heading
[constr_3055]	SystemSignalGroup in a complete System Description
[constr_3059]	Mandatory DataMapping on the receiver side for elements of a composite data type
[constr_3063]	Usage of portNumber and dynamicallyAssigned with value “true” is mutually exclusive
[constr_3064]	Usage of serviceInstance , eventHandler and eventGroup references
[constr_3087]	DataMapping to PRPortPrototype
[constr_3088]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3176]	Value range of windowSize
[constr_3177]	Dependency between maxErrorStateValid , maxErrorStateInit and maxErrorStateInvalid
[constr_3178]	Dependency between minOkStateValid , minOkStateInit and minOkStateInvalid
[constr_3179]	Dependency between minOkStateInit , maxErrorStateInit and windowSize
[constr_3180]	Dependency between minOkStateValid , maxErrorStateValid and windowSize
[constr_3181]	Dependency between minOkStateInvalid , maxErrorStateInvalid and windowSize
[constr_3201]	eventGroupIdentifier in ConsumedEventGroups that are referenced by the same EventHandler
[constr_3259]	Allowed use of SdServerConfig.capabilityRecord
[constr_3260]	Allowed use of SdClientConfig.capabilityRecord
[constr_3271]	clientIpAddrFromConnectionRequest and clientPortFromConnectionRequest settings for SD SocketConnections
[constr_3400]	Usage of SdClientConfig attributes in ConsumedServiceInstance and ConsumedEventGroup
[constr_3401]	Usage of SdServerConfig attributes in ProvidedServiceInstance and EventHandler

Table D.53: Deleted Constraints in 19-11

E Mentioned Class Tables

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	ARElement (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet , ApplicationPartition , AutosarData Type , Base Type , BlueprintMappingSet, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet , ClientServerInterfaceToBswModuleEntryBlueprintMapping, Collection, CompuMethod , ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CryptoServiceCertificate , CryptoServiceKey , CryptoServicePrimitive , CryptoServiceQueue , DataConstr, DataExchangePoint, DataTransformationSet , DataTypeMappingSet , DiagnosticCommonElement , DiagnosticConnection , DiagnosticContributionSet, DiagnosticMasterToSlaveEventMappingSet, Documentation, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EndToEndProtectionSet , EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FlatMap , GeneralPurposeConnection , HwCategory, HwElement , HwType, IPSecConfigProps , IPv6ExtHeaderFilterSet , Implementation , InterpolationRoutineMappingSet, J1939ControllerApplication , KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, McFunction, McGroup, ModeDeclarationGroup , ModeDeclarationMappingSet, PhysicalDimension, PhysicalDimensionMappingSet, PortInterface , PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, SignalServiceTranslationPropsSet , SomeipSdClientEventGroupTimingConfig , SomeipSdClientServiceInstanceConfig , SomeipSdServerEventGroupTimingConfig , SomeipSdServerServiceInstanceConfig , SwAddrMethod, SwAxisType, SwComponentType , SwRecordLayout , SwSystemconst , SwSystemconstantValueSet , SwcBswMapping, System, SystemSignal, SystemSignalGroup, TcpOptionFilterSet, TimingExtension , TlvDataIdDefinitionSet , TransformationPropsSet , Unit, UnitGroup, ViewMapSet			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.1: ARElement

Class	ARPackage			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	AUTOSAR package, allowing to create top level packages to structure the contained ARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM.			
Base	<i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–





Class	ARPackage			
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	aggr	This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. Stereotypes: atpSplittable Tags: atp.Splitkey=shortLabel xml.sequenceOffset=10

Table E.2: ARPackage

Class	AdminData			
Package	M2::MSR::AsamHdo::AdminData			
Note	AdminData represents the ability to express administrative information for an element. This administration information is to be treated as meta-data such as revision id or state of the file. There are basically four kinds of meta-data <ul style="list-style-type: none"> • The language and/or used languages. • Revision information covering e.g. revision number, state, release date, changes. Note that this information can be given in general as well as related to a particular company. • Document meta-data specific for a company 			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
docRevision (ordered)	DocRevision	*	aggr	This allows to denote information about the current revision of the object. Note that information about previous revisions can also be logged here. The entries shall be sorted descendant by date in order to reflect the history. Therefore the most recent entry representing the current version is denoted first. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=50 xml.typeElement=false xml.typeWrapperElement=false





Class	AdminData			
language	LEnum	0..1	attr	This attribute specifies the master language of the document or the document fragment. The master language is the one in which the document is maintained and from which the other languages are derived from. In particular in case of inconsistencies, the information in the master language is priority. Tags: xml.sequenceOffset=20
sdg	Sdg	*	aggr	This property allows to keep special data which is not represented by the standard model. It can be utilized to keep e.g. tool specific data. Stereotypes: atpSplitable Tags: atp.Splitkey=sdg xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=60 xml.typeElement=false xml.typeWrapperElement=false
usedLanguages	MultiLanguagePlainText	0..1	aggr	This property specifies the languages which are provided in the document. Therefore it should only be specified in the top level admin data. For each language provided in the document there is one entry in MultilanguagePlainText. The content of each entry can be used for illustration of the language. The used language itself depends on the language attribute in the entry. Tags: xml.sequenceOffset=30

Table E.3: AdminData

Class	AnyInstanceRef			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::AnyInstanceRef			
Note	Describes a reference to any instance in an AUTOSAR model. This is the most generic form of an instance ref. Refer to the superclass notes for more details.			
Base	<i>ARObject</i> , <i>AtpInstanceRef</i>			
Attribute	Type	Mult.	Kind	Note
base	AtpClassifier	1	ref	This is the base from which navigation path begins. Stereotypes: atpDerived
contextElement	AtpFeature	*	ref	This is one step in the navigation path specified by the instance ref.
target	AtpFeature	1	ref	This is the target of the instance ref.

Table E.4: AnyInstanceRef

Class	ApplicationArrayDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which is an array, each element is of the same application data type. Tags: atp.recommendedPackage=ApplicationDataTypes			





Class	ApplicationArrayDataType			
Base	ARElement , ARObject , ApplicationCompositeDataType , ApplicationDataType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.
element	ApplicationArrayElement	1	aggr	This association implements the concept of an array element. That is, in some cases it is necessary to be able to identify single array elements, e.g. as input values for an interpolation routine.

Table E.5: ApplicationArrayDataType

Class	ApplicationCompositeDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for all application data types composed of other data types.			
Base	ARElement , ARObject , ApplicationDataType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	ApplicationArrayDataType , ApplicationRecordDataType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.6: ApplicationCompositeDataType

Class	ApplicationCompositeElementDataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	This class represents a data prototype which is aggregated within a composite application data type (record or array). It is introduced to provide a better distinction between target and context in instance Refs.			
Base	ARObject , AtpFeature , AtpPrototype , DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	ApplicationArrayElement, ApplicationRecordElement			
Attribute	Type	Mult.	Kind	Note
type	ApplicationDataType	1	tref	This represents the corresponding data type. Stereotypes: isOfType

Table E.7: ApplicationCompositeElementDataPrototype

Class	ApplicationDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	ApplicationDataType defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An ApplicationDataType represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianness, etc. It should be possible to model the application level aspects of a VFB system by using ApplicationDataTypes only.			





Class	ApplicationDataType (abstract)			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	ApplicationCompositeDataType , ApplicationPrimitiveDataType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.8: ApplicationDataType

Class	ApplicationPrimitiveDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	A primitive data type defines a set of allowed values. Tags: atp.recommendedPackage=ApplicationDataTypes			
Base	ARElement , ARObject , ApplicationDataType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.9: ApplicationPrimitiveDataType

Class	ApplicationRecordDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which can be decomposed into prototypes of other application data types. Tags: atp.recommendedPackage=ApplicationDataTypes			
Base	ARElement , ARObject , ApplicationCompositeDataType , ApplicationDataType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , AutosarDataType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
element (ordered)	ApplicationRecordElement	1..*	aggr	Specifies an element of a record. The aggregation of ApplicationRecordElement is subject to variability with the purpose to support the conditional existence of elements inside a ApplicationrecordData Type. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.10: ApplicationRecordDataType

Class	ApplicationRecordElement			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Describes the properties of one particular element of an application record data type.			
Base	ARObject , ApplicationCompositeElementDataPrototype , AtpFeature , AtpPrototype , DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–





Class	ApplicationRecordElement			
isOptional	Boolean	0..1	attr	<p>This attribute represents the ability to declare the enclosing ApplicationRecordElement as optional. This means the that, at runtime, the ApplicationRecordElement may or may not have a valid value and shall therefore be ignored.</p> <p>The underlying runtime software provides means to set the ApplicationRecordElement as not valid at the sending end of a communication and determine its validity at the receiving end.</p>

Table E.11: ApplicationRecordElement

Class	ApplicationSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	<p>The ApplicationSwComponentType is used to represent the application software.</p> <p>Tags:atp.recommendedPackage=SwComponentTypes</p>			
Base	ARElement , ARObject , AtomicSwComponentType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.12: ApplicationSwComponentType

Class	ArgumentDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	<p>An argument of an operation, much like a data element, but also carries direction information and is owned by a particular ClientServerOperation.</p>			
Base	ARObject , AtpFeature , AtpPrototype , AutosarDataPrototype , DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
direction	ArgumentDirectionEnum	1	attr	This attribute specifies the direction of the argument prototype.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	0..1	attr	<p>This defines how the argument type of the servers RunnableEntity is implemented.</p> <p>If the attribute is not defined this has the same semantics as if the attribute is set to the value useArgumentType for primitive arguments and structures.</p>

Table E.13: ArgumentDataPrototype

Enumeration	ArgumentDirectionEnum			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes			
Note	<p>Use cases:</p> <ul style="list-style-type: none"> Arguments in ClientServerOperation can have different directions that need to be formally indicated because they have an impact on how the function signature looks like eventually. Arguments in BswModuleEntry already determine a function signature, but the direction is used to specify the semantics, especially of pointer arguments. 			





Enumeration	ArgumentDirectionEnum
Literal	Description
in	The argument value is passed to the callee. Tags: atp.EnumerationLiteralIndex=0
inout	The argument value is passed to the callee but also passed back from the callee to the caller. Tags: atp.EnumerationLiteralIndex=1
out	The argument value is passed from the callee to the caller. Tags: atp.EnumerationLiteralIndex=2

Table E.14: ArgumentDirectionEnum

Class	ArrayValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specifies the values for an array.			
Base	<i>ARObject</i> , <i>CompositeValueSpecification</i> , <i>ValueSpecification</i>			
Attribute	Type	Mult.	Kind	Note
element (ordered)	ValueSpecification	*	aggr	The value for a single array element. All Value Specifications aggregated by ArrayValueSpecification shall have the same structure. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
intendedPartialInitializationCount	PositiveInteger	0..1	attr	This attribute shall only have a meaning for dynamic arrays and shall be taken as a sanity check: the number filled in the attribute shall be identical to the number of ArrayValueSpecification.element. If the attribute does not exist it means that no partial initialization is intended.

Table E.15: ArrayValueSpecification

Class	AssemblySwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	AssemblySwConnectors are exclusively used to connect SwComponentPrototypes in the context of a CompositionSwComponentType.			
Base	<i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , Identifiable , MultilanguageReferrable , Referrable , SwConnector			
Attribute	Type	Mult.	Kind	Note
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port.
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port.

Table E.16: AssemblySwConnector

Class	AtomicSwComponentType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Subclasses	ApplicationSwComponentType , ComplexDeviceDriverSwComponentType , EcuAbstractionSwComponentType , NvBlockSwComponentType , SensorActuatorSwComponentType , ServiceProxySwComponentType , ServiceSwComponentType			
Attribute	Type	Mult.	Kind	Note
internalBehavior	SwcInternalBehavior	0..1	aggr	The SwcInternalBehaviors owned by an AtomicSw ComponentType can be located in a different physical file. Therefore the aggregation is <<atpSplitable>>. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=internalBehavior, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSw ComponentType. Stereotypes: atpSplitable Tags: atp.Splitkey=shortName

Table E.17: AtomicSwComponentType

Class	AtpInstanceRef (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node. An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).			
Base	ARObject			
Subclasses	AnyInstanceRef , ApplicationCompositeElementInPortInterfaceInstanceRef , ComponentInCompositionInstanceRef , ComponentInSystemInstanceRef , DataPrototypeInPortInterfaceInstanceRef , DataPrototypeInSystemInstanceRef , InnerDataPrototypeGroupInCompositionInstanceRef , InnerPortGroupInCompositionInstanceRef , InnerRunnableEntityGroupInCompositionInstanceRef , InstanceEventInCompositionInstanceRef , ModeGroupInAtomicSwcInstanceRef , ModelInBswModuleDescriptionInstanceRef , ModelInSwcInstanceRef , OperationArgumentInComponentInstanceRef , OperationInAtomicSwcInstanceRef , OperationInSystemInstanceRef , PModelInSystemInstanceRef , ParameterInAtomicSWCTypeInstanceRef , PortGroupInSystemInstanceRef , PortInCompositionTypeInstanceRef , RModelInAtomicSwcInstanceRef , RteEventInEcuInstanceRef , RunnableEntityInCompositionInstanceRef , SwcServiceDependencyInSystemInstanceRef , TriggerInAtomicSwcInstanceRef , TriggerInSystemInstanceRef , VariableAccessInEcuInstanceRef , VariableDataPrototypeInCompositionInstanceRef , VariableDataPrototypeInSystemInstanceRef , VariableInAtomicSWCTypeInstanceRef , VariableInAtomicSwcInstanceRef , VariableInComponentInstanceRef			
Attribute	Type	Mult.	Kind	Note
atpBase	AtpClassifier	1	ref	This is the base from which the navigaion path starts. Stereotypes: atpAbstract; atpDerived
atpContext Element (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. Stereotypes: atpAbstract





Class	AtpInstanceRef (abstract)			
atpTarget	AtpFeature	1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. Stereotypes: atpAbstract

Table E.18: AtpInstanceRef

Class	AutosarDataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of an AutosarDataType.			
Base	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>DataPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>ArgumentDataPrototype</i> , <i>ParameterDataPrototype</i> , <i>VariableDataPrototype</i>			
Attribute	Type	Mult.	Kind	Note
type	AutosarDataType	1	tref	This represents the corresponding data type. Stereotypes: isOfType

Table E.19: AutosarDataPrototype

Class	AutosarDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for user defined AUTOSAR data types for ECU software.			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	<i>AbstractImplementationDataType</i> , <i>ApplicationDataType</i>			
Attribute	Type	Mult.	Kind	Note
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType.

Table E.20: AutosarDataType

Class	BaseType (abstract)			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependant base type.			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	SwBaseType			
Attribute	Type	Mult.	Kind	Note
baseType Definition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table E.21: BaseType

Class	BaseTypeDirectDefinition			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	ARObject, BaseTypeDefinition			
Attribute	Type	Mult.	Kind	Note
baseTypeEncoding	BaseTypeEncodingString	1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. Tags: xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. Tags: xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. Tags: xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". Tags: xml.sequenceOffset=100
nativeDeclaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE. If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size. This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems. Tags: xml.sequenceOffset=120

Table E.22: BaseTypeDirectDefinition

Class	BswInternalBehavior			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies the behavior of a BSW module or a BSW cluster w.r.t. the code entities visible by the BSW Scheduler. It is possible to have several different BswInternalBehaviors referring to the same BswModule Description.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , InternalBehavior , Multilanguage Referrable , Referrable			
Attribute	Type	Mult.	Kind	Note





Class	BswInternalBehavior			
arTypedPerInstanceMemory	VariableDataPrototype	*	aggr	<p>Defines an AUTOSAR typed memory-block that needs to be available for each instance of the Basic Software Module. The aggregation of arTypedPerInstanceMemory is subject to variability with the purpose to support variability in the Basic Software Module's implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
bswPerInstanceMemoryPolicy	BswPerInstanceMemoryPolicy	*	aggr	<p>Policy for a arTypedPerInstanceMemory The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=bswPerInstanceMemoryPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
clientPolicy	BswClientPolicy	*	aggr	<p>Policy for a requiredClientServerEntry. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=clientPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
distinguishedPartition	BswDistinguishedPartition	*	aggr	<p>Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60</p>
entity	BswModuleEntity	*	aggr	<p>A code entity for which the behavior is described</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=5</p>
event	BswEvent	*	aggr	<p>An event required by this module behavior.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=10</p>
exclusiveAreaPolicy	BswExclusiveAreaPolicy	*	aggr	<p>Policy for an ExclusiveArea in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=exclusiveAreaPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>





Class	BswInternalBehavior			
includedData TypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a basic software module for its implementation. Stereotypes: atpSplitable Tags: atp.Splitkey=includedDataTypeSet
internal TriggeringPoint	BswInternalTriggering Point	*	aggr	An internal triggering point. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=2
internal TriggeringPoint Policy	BswInternalTriggering PointPolicy	*	aggr	Policy for an internalTriggeringPoint in this BswInternal Behavior.. The policy selects the options of the Schedule Manager API generation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=internalTriggeringPointPolicy, variation Point.shortLabel vh.latestBindingTime=preCompileTime
modeReceiver Policy	BswModeReceiver Policy	*	aggr	Implementation policy for the reception of mode switches. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=modeReceiverPolicy, variationPoint.short Label vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
modeSender Policy	BswModeSenderPolicy	*	aggr	Implementation policy for providing a mode group. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=modeSenderPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
parameterPolicy	BswParameterPolicy	*	aggr	Policy for a perInstanceParameter in this BswInternal Behavior. The policy selects the options of the Schedule Manager API generation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=parameterPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
perInstance Parameter	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) needed by this BswInternal Behavior. The role name perInstanceParameter is chosen in analogy to the similar role in the context of SwcInternal Behavior. In contrast to constantMemory, this object is not allocated locally by the module's code, but by the BSW Scheduler and it is accessed from the BSW module via the BSW Scheduler API. The main use case is the support of software emulation of calibration data. The aggregation is subject to variability with the purpose to support implementation variants. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45





Class	BswInternalBehavior			
receptionPolicy	BswDataReception Policy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=receptionPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
releasedTrigger Policy	BswReleasedTrigger Policy	*	aggr	Policy for a releasedTrigger. The policy selects the options of the Schedule Manager API generation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=releasedTriggerPolicy, variationPoint.short Label vh.latestBindingTime=preCompileTime
schedulerName Prefix	BswSchedulerName Prefix	*	aggr	Optional definition of one or more prefixes to be used for the BswScheduler. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50
sendPolicy	BswDataSendPolicy	*	aggr	Policy for a providedData. The policy selects the options of the Schedule Manager API generation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=sendPolicy, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
service Dependency	BswService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item. The aggregation is subject to variability with the purpose to support the conditional existence of ServiceNeeds. The aggregation is splitable in order to support that ServiceNeeds might be provided in later development steps. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=ident.shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
triggerDirect Implementation	BswTriggerDirect Implementation	*	aggr	Specifies a trigger to be directly implemented via OS calls. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=triggerDirectImplementation, variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=15
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation. Stereotypes: atpSplitable Tags: atp.Splitkey=shortName

Table E.23: BswInternalBehavior

Class	CalibrationParameterValue			
Package	M2::AUTOSARTemplates::SWComponentTemplate::MeasurementAndCalibration::CalibrationParameter Values			
Note	<p>Specifies instance specific calibration parameter values used to initialize the memory objects implementing calibration parameters in the generated RTE code.</p> <p>RTE generator will use the implInitValue to override the initial values specified for the DataPrototypes of a component type.</p> <p>The applInitValue is used to exchange init values with the component vendor not publishing the transformation algorithm between ApplicationDataTypes and ImplementationDataTypes or defining an instance specific initialization of components which are only defined with ApplicationDataTypes.</p> <p>Note: If both representations of init values are available these need to represent the same content.</p> <p>Note further that in this case an explicit mapping of ValueSpecification is not implemented because calibration parameters are delivered back after the calibration phase.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
applInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ApplicationDataType
implInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ImplementationDataType
initialized Parameter	FlatInstanceDescriptor	1	ref	This represents the parameter that is initialized by the CalibrationParameterValue.

Table E.24: CalibrationParameterValue

Class	ClientServerInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	<p>A client/server interface declares a number of operations that can be invoked on a server by a client.</p> <p>Tags:atp.recommendedPackage=PortInterfaces</p>			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , PortInterface , Referrable			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	1..*	aggr	<p>ClientServerOperation(s) of this ClientServerInterface.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=blueprintDerivationTime</p>
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

Table E.25: ClientServerInterface

Class	ClientServerOperation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	An operation declared within the scope of a client/server interface.			
Base	ARObject , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
argument (ordered)	ArgumentDataPrototype	*	aggr	<p>An argument of this ClientServerOperation</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=blueprintDerivationTime</p>





Class	ClientServerOperation			
diagArgIntegrity	Boolean	0..1	attr	<p>This attribute shall only be used in the implementation of diagnostic routines to support the case where input and output arguments are allocated in a shared buffer and might unintentionally overwrite input arguments by tentative write operations to output arguments.</p> <p>This situation can happen during sliced execution or while output parameters are arrays (call by reference). The value true means that the ClientServerOperation is aware of the usage of a shared buffer and takes precautions to avoid unintentional overwrite of input arguments.</p> <p>If the attribute does not exist or is set to false the Client ServerOperation does not have to consider the usage of a shared buffer.</p>
possibleError	ApplicationError	*	ref	Possible errors that may be raised by the referring operation.

Table E.26: ClientServerOperation

Class	CompositionSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	<p>A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by SwComponentTypes) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means hierarchical structures of software-components can be created.</p> <p>Tags:atp.recommendedPackage=SwComponentTypes</p>			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Attribute	Type	Mult.	Kind	Note
component	SwComponentPrototype	*	aggr	<p>The instantiated components that are part of this composition. The aggregation of SwComponentPrototype is subject to variability with the purpose to support the conditional existence of a SwComponentPrototype. Please be aware: if the conditional existence of SwComponentPrototypes is resolved post-build the deselected SwComponentPrototypes are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE.</p> <p>The aggregation is marked as atpSplitable in order to allow the addition of service components to the ECU extract during the ECU integration.</p> <p>The use case for having 0 components owned by the CompositionSwComponentType could be to deliver an empty CompositionSwComponentType to e.g. a supplier for filling the internal structure.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>





Class	CompositionSwComponentType			
connector	SwConnector	*	aggr	<p>SwConnectors have the principal ability to establish a connection among PortPrototypes. They can have many roles in the context of a CompositionSwComponentType. Details are refined by subclasses.</p> <p>The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow.</p> <p>The aggregation is marked as atpSplitable in order to allow the extension of the ECU extract with AssemblySw Connectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
constantValue Mapping	ConstantSpecification MappingSet	*	ref	<p>Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortCom Spec.</p> <p>Stereotypes: atpSplitable Tags: atp.Splitkey=constantValueMapping</p>
dataType Mapping	DataTypeMappingSet	*	ref	<p>Reference to the DataTypeMapping to be applied for the used ApplicationDataTypes in PortInterfaces.</p> <p>Background: when developing subsystems it may happen that ApplicationDataTypes are used on the surface of CompositionSwComponentTypes. In this case it would be reasonable to be able to also provide the intended mapping to the ImplementationDataTypes. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the CompositionSwComponent Types.</p> <p>Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility.</p> <p>Stereotypes: atpSplitable Tags: atp.Splitkey=dataTypeMapping</p>
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	<p>This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortLabel, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p>

Table E.27: CompositionSwComponentType

Class	CompuMethod
Package	M2::MSR::AsamHdo::ComputationMethod
Note	<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.</p> <p>Tags: atp.recommendedPackage=CompuMethods</p>





Class		CompuMethod		
Base	<i>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
compuInternalToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. Tags: xml.sequenceOffset=80
compuPhysToInternal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. Tags: xml.sequenceOffset=90
displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. Tags: xml.sequenceOffset=20
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. Tags: xml.sequenceOffset=30

Table E.28: CompuMethod

Class		CompuScale		
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to specify one segment of a segmented computation method.			
Base	<i>ARObject</i>			
Attribute	Type	Mult.	Kind	Note
compuInverseValue	CompuConst	0..1	aggr	This is the inverse value of the constraint. This supports the case that the scale is not reversible per se. Tags: xml.sequenceOffset=60
compuScaleContents	CompuScaleContents	0..1	aggr	This represents the computation details of the scale. Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=70 xml.typeElement=false xml.typeWrapperElement=false
desc	MultiLanguageOverviewParagraph	0..1	aggr	<desc> represents a general but brief description of the object in question. Tags: xml.sequenceOffset=30
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the scale. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=40





Class	CompuScale			
mask	PositiveInteger	0..1	attr	<p>In difference to all the other computational methods every COMPU-SCALE will be applied including the bit MASK. Therefore it is allowed for this type of COMPU-METHOD, that COMPU-SCALES overlap.</p> <p>To calculate the string reverse to a value, the string has to be split and the according value for each substring has to be summed up. The sum is finally transmitted.</p> <p>The processing has to be done in order of the COMPU-SCALE elements.</p> <p>Tags:xml.sequenceOffset=35</p>
shortLabel	Identifier	0..1	attr	<p>This element specifies a short name for the particular scale. The name can for example be used to derive a programming language identifier.</p> <p>Tags:xml.sequenceOffset=20</p>
symbol	CIdentifier	0..1	attr	<p>The symbol, if provided, is used by code generators to get a C identifier for the CompuScale. The name will be used as is for the code generation, therefore it needs to be unique within the generation context.</p> <p>Tags:xml.sequenceOffset=25</p>
upperLimit	Limit	0..1	attr	<p>This specifies the upper limit of a of the scale.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=50</p>

Table E.29: CompuScale

Class	<<atpMixedString>> ConditionByFormula			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	<p>This class represents a condition which is computed based on system constants according to the specified expression. The expected result is considered as boolean value.</p> <p>The result of the expression is interpreted as a condition.</p> <ul style="list-style-type: none"> • "0" represents "false"; • a value other than zero is considered "true" 			
Base	<i>ARObject, FormulaExpression, SwSystemconstDependentFormula</i>			
Attribute	Type	Mult.	Kind	Note
bindingTime	BindingTimeEnum	1	attr	<p>This attribute specifies the point in time when condition may be evaluated at earliest. At this point in time all referenced system constants shall have a value.</p> <p>Tags:xml.attribute=true</p>

Table E.30: ConditionByFormula

Enumeration	CouplingPortRoleEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines the role a CouplingPort takes in the context of a CouplingElement.
Literal	Description





Enumeration	CouplingPortRoleEnum
hostPort	The hostPort is connected to an ECU (host ecu). The host ECU controls the connected Coupling Element (e.g. Ethernet switch). Tags: atp.EnumerationLiteralIndex=0
standardPort	A CouplingPort can be a standardPort that is used to connect the CouplingElement with Coupling Ports outside the ECU. Tags: atp.EnumerationLiteralIndex=2
upLinkPort	A CouplingPort can be connected to another CouplingPort of a CouplingElement located on the same ECU (CouplingElement.ecuInstance) using the CouplingPortConnection. This is used to model a cascaded switch. Tags: atp.EnumerationLiteralIndex=1

Table E.31: CouplingPortRoleEnum

Class	DataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of any data type.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ApplicationCompositeElementDataPrototype, AutosarDataPrototype			
Attribute	Type	Mult.	Kind	Note
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

Table E.32: DataPrototype

Class	DataPrototypeMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	<p>Defines the mapping of two particular VariableDataPrototypes, ParameterDataPrototypes or Argument DataPrototypes with unequal names and/or unequal semantic (resolution or range) in context of two different SenderReceiverInterface, NvDataInterface or ParameterInterface or Operations.</p> <p>If the semantic is unequal following rules apply: The textTableMapping is only applicable if the referred DataPrototypes are typed by AutosarDataType referring to CompuMethods of category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE or BITFIELD_TEXTTABLE.</p> <p>In the case that the DataPrototypes are typed by AutosarDataType either referring to CompuMethods of category LINEAR, IDENTICAL or referring to no CompuMethod (which is similar as IDENTICAL) the linear conversion factor is calculated out of the factorSiToUnit and offsetSiToUnit attributes of the referred Units and the CompuRationalCoeffs of a compuInternalToPhys of the referred CompuMethods.</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
firstData Prototype	AutosarDataPrototype	1	ref	First to be mapped DataPrototype in context of a Sender ReceiverInterface, NvDataInterface, ParameterInterface or Operation.
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference defines the need to execute the Data Transformation <Mip>_<transformerId> functions of the transformation chain when communicating from the Data PrototypeMapping.firstDataPrototype to the Data PrototypeMapping.secondDataPrototype.





Class	DataPrototypeMapping			
				<p>△</p> <p>This reference also specifies the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain (i.e. from the DataPrototypeMapping.secondDataPrototype to the DataPrototypeMapping.firstDataPrototype) if the referenced Data Transformation is symmetric, i.e. attribute DataTransformation.dataTransformationKind is set to symmetric.</p>
secondDataPrototype	AutosarDataPrototype	1	ref	Second to be mapped DataPrototype in context of a SenderReceiverInterface, NvDataInterface, ParameterInterface or Operation.
secondToFirstDataTransformation	DataTransformation	0..1	ref	This defines the need to execute the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain when communicating from the DataPrototypeMapping.secondDataPrototype to the DataPrototypeMapping.firstDataPrototype.
subElementMapping	SubElementMapping	*	aggr	This represents the owned SubelementMapping.
textTableMapping	TextTableMapping	0..2	aggr	Applied TextTableMapping(s)

Table E.33: DataPrototypeMapping

Enumeration	DataTransformationErrorHandlingEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions
Note	This enumeration defines different ways how a RunnableEntity shall handle transformer errors.
Literal	Description
noTransformerErrorHandling	A runnable does not handle transformer errors. Tags: atp.EnumerationLiteralIndex=0
transformerErrorHandling	The runnable implements the handling of transformer errors. Tags: atp.EnumerationLiteralIndex=1

Table E.34: DataTransformationErrorHandlingEnum

Enumeration	DataTransformationStatusForwardingEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions
Note	This enumeration defines different ways how a RunnableEntity shall be able to forward status into the transformer chain. Tags: atp.Status=draft
Literal	Description
noTransformerStatusForwarding	The runnable is not able to forward a transformer status. Tags: atp.EnumerationLiteralIndex=0
transformerStatusForwarding	The runnable is able to forward a transformer status. Tags: atp.EnumerationLiteralIndex=1

Table E.35: DataTransformationStatusForwardingEnum

Class	DataTypeMap			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	This class represents the relationship between ApplicationDataType and its implementing AbstractImplementationDataType.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
applicationDataType	ApplicationDataType	1	ref	This is the corresponding ApplicationDataType
implementationDataType	AbstractImplementationDataType	1	ref	This is the corresponding AbstractImplementationDataType.

Table E.36: DataTypeMap

Class	DataTypeMappingSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	This class represents a list of mappings between ApplicationDataTypes and ImplementationDataTypes. In addition, it can contain mappings between ImplementationDataTypes and ModeDeclarationGroups. Tags: atp.recommendedPackage=DataTypeMappingSets			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mult.	Kind	Note
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an ApplicationDataType and its AbstractImplementationDataType.
modeRequestTypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an ModeDeclarationGroup and its AbstractImplementationDataType.

Table E.37: DataTypeMappingSet

Class	DelegationSwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	A delegation connector delegates one inner PortPrototype (a port of a component that is used inside the composition) to a outer PortPrototype of compatible type that belongs directly to the composition (a port that is owned by the composition).			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, SwConnector			
Attribute	Type	Mult.	Kind	Note
innerPort	PortPrototype	1	iref	The port that belongs to the ComponentPrototype in the composition Tags: xml.typeElement=true
outerPort	PortPrototype	1	ref	The port that is located on the outside of the Composition Type

Table E.38: DelegationSwConnector

Class	Describable (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	This meta-class represents the ability to add a descriptive documentation to non identifiable elements.			
Base	<i>ARObject</i>			
Subclasses	CyclicTiming , EventControlledTiming , HwElementConnector , HwPinConnector , HwPinGroupConnector , I PduTiming , Ipv4DhcpServerConfiguration , Ipv6DhcpServerConfiguration , PncMapping , Socket Connection , TransformationComSpecProps , TransformationDescription , TransformationISignalProps			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data for the describable object. Tags: xml.sequenceOffset=-20
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Describable. It affects the expected existence of attributes and the applicability of constraints. Tags: xml.sequenceOffset=-50
desc	MultiLanguageOverview Paragraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". Tags: xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. Tags: xml.sequenceOffset=-30

Table E.39: Describable

Class	DolpActivationLineNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	A DoIP entity needs to be informed when an external tester is attached or activated. The DolpActivation ServiceNeeds specifies the trigger for such an event. Examples would be a Pdu via a regular communication bus, a PWM signal, or an I/O. For details please refer to the ISO 13400.			
Base	<i>ARObject</i> , DolpServiceNeeds , Identifiable , MultilanguageReferrable , Referrable , ServiceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.40: DolpActivationLineNeeds

Class	DolpGidNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DolpGidNeeds indicates that the software-component owning this ServiceNeeds is providing the GID number either after a GID Synchronisation or by other means like e.g. flashed EEPROM parameter. This need can be used independent from DolpGidSynchronizationNeeds and is necessary if the GID can not be provided out of the DoIP configuration options.			
Base	<i>ARObject</i> , DolpServiceNeeds , Identifiable , MultilanguageReferrable , Referrable , ServiceNeeds			





Class	DolpGidNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.41: DolpGidNeeds

Class	DolpGidSynchronizationNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DolpGidSynchronizationNeeds indicates that the software-component owning this ServiceNeeds is triggered by the DoIP entity to start a synchronization of the GID (Group Identification) on the DoIP service 0x0001, 0x0002, 0x0003 or before announcement via service 0x0004 according to ISO 13400-2:2012 if necessary. Note that this need is only relevant for DoIP synchronization masters.			
Base	<i>ARObject</i> , <i>DolpServiceNeeds</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>ServiceNeeds</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.42: DolpGidSynchronizationNeeds

Class	DolpPowerModeStatusNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DolpPowerModeStatusNeeds indicates that the software-component owning this ServiceNeeds is providing the PowerModeStatus for the DoIP service 0x4003 according to ISO 13400-2:2012.			
Base	<i>ARObject</i> , <i>DolpServiceNeeds</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>ServiceNeeds</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.43: DolpPowerModeStatusNeeds

Class	DolpServiceNeeds (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This represents an abstract base class for ServiceNeeds related to DoIP.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>ServiceNeeds</i>			
Subclasses	<i>DolpActivationLineNeeds</i> , <i>DolpGidNeeds</i> , <i>DolpGidSynchronizationNeeds</i> , <i>DolpPowerModeStatusNeeds</i> , <i>DolpRoutingActivationAuthenticationNeeds</i> , <i>DolpRoutingActivationConfirmationNeeds</i> , <i>FurtherActionByteNeeds</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.44: DolpServiceNeeds

Class	EcucContainerDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Base class used to gather common attributes of configuration container definitions.			
Base	<i>ARObject</i> , <i>AtpDefinition</i> , <i>EcucDefinitionElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			





Class	<i>EcucContainerDef</i> (abstract)			
Subclasses	EcucChoiceContainerDef, EcucParamConfContainerDef			
Attribute	Type	Mult.	Kind	Note
destinationUri	EcucDestinationUriDef	*	ref	Several destinationUris can be defined for an Ecuc ContainerDef. With such destinationUris an Ecuc ContainerDef is applicable for several EcucUriReference Defs. Stereotypes: atpUriDef
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies which MultiplicityConfigurationClass this container is available for which ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModule Def is set to VENDOR_SPECIFIC_MODULE_DEFINITION and if the upperMultiplicity is greater than the lowerMultiplicity then this aggregation is mandatory. Tags: xml.name Plural=MULTIPLICITY-CONFIG-CLASSES
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a container may have different number of instances in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
requiresIndex	Boolean	0..1	attr	Used to define whether the value element for this definition shall be provided with an index.

Table E.45: EcucContainerDef

Class	EcucContainerValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Represents a Container definition in the ECU Configuration Description.			
Base	<i>ARObject</i> , <i>EcucIndexableValue</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
definition	EcucContainerDef	1	ref	Reference to the definition of this Container in the ECU Configuration Parameter Definition. Tags: xml.sequenceOffset=-10
parameterValue	EcucParameterValue	*	aggr	Aggregates all ECU Configuration Values within this Container. atpVariation: [RS_ECUC_00079] Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=definition, variationPoint.shortLabel vh.latestBindingTime=postBuild
referenceValue	EcucAbstractReference Value	*	aggr	Aggregates all References with this container. atpVariation: [RS_ECUC_00079] Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=definition, variationPoint.shortLabel vh.latestBindingTime=postBuild





Class	EcucContainerValue			
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container. atpVariation: [RS_ECUC_00078] Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=definition, shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table E.46: EcucContainerValue

Class	EndToEndProtectionVariablePrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	It is possible to protect the data exchanged between software components. For this purpose, for each communication to be protected, the user defines a separate EndToEndProtection (specifying a set of protection settings) and refers to a variableDataPrototype in the role of sender and to one or many variableDataPrototypes in the role of receiver. For details, see EndToEnd Library.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
receiver	VariableDataPrototype	*	iref	This represents the receiver. Note that 1:n communication is supported for this use case.
sender	VariableDataPrototype	0..1	iref	This represents the sender. Can be optional if an ecu extract is provided and the sender is part of the extract.
shortLabel	Identifier	0..1	attr	This serves as part of the split key in case of more than one EndToEndProtectionVariablePrototype is aggregated in the bound model.

Table E.47: EndToEndProtectionVariablePrototype

Class	EndToEndTransformationComSpecProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class EndToEndTransformationComSpecProps specifies port specific configuration properties for EndToEnd transformer attributes.			
Base	ARObject, Describable , TransformationComSpecProps			
Attribute	Type	Mult.	Kind	Note
clearFromValidToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
disableEndToEndCheck	Boolean	1	attr	Disables/Enables the E2E check. The E2Eheader is removed from the payload independent from the setting of this attribute.
maxDeltaCounter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and Max DeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorStateInit	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INIT. The minimum value is 0.





Class	EndToEndTransformationComSpecProps			
maxErrorState Invalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INVALID. The minimum value is 0.
maxErrorState Valid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_VALID. The minimum value is 0.
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT. The minimum value is 1.
minOkState Invalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID. The minimum value is 1.
minOkState Valid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID. The minimum value is 1.
syncCounterInit	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.
windowSize Invalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSize Valid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table E.48: EndToEndTransformationComSpecProps

Class	ExecutableEntity (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Abstraction of executable code.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	BswModuleEntity , RunnableEntity			
Attribute	Type	Mult.	Kind	Note
activation Reason	ExecutableEntity ActivationReason	*	aggr	If the ExecutableEntity provides at least one activation Reason element the RTE resp. BSW Scheduler shall provide means to read the activation vector of this executable entity execution. If no activationReason element is provided the feature of being able to determine the activating RTEEvent is disabled for this ExecutableEntity.
canEnter ExclusiveArea	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls.





Class	<i>ExecutableEntity</i> (abstract)			
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.
minimumStartInterval	TimeValue	1	attr	Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.
reentrancyLevel	ReentrancyLevelEnum	0..1	attr	The reentrancy level of this ExecutableEntity. See the documentation of the enumeration type ReentrancyLevelEnum for details. Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.
runsInsideExclusiveArea	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area.
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this code entity. Via an association to the same SwAddrMethod, it can be specified that several code entities (even of different modules or components) shall be located in the same memory without already specifying the memory section itself.

Table E.49: ExecutableEntity

Class	<i>ExecutionTime</i> (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Base class for several means how to describe the ExecutionTime of software. The required context information is provided through this class.			
Base	<i>ARObject</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Subclasses	AnalyzedExecutionTime, MeasuredExecutionTime, RoughEstimateOfExecutionTime, SimulatedExecutionTime			
Attribute	Type	Mult.	Kind	Note
exclusiveArea	ExclusiveArea	0..1	ref	Reference to the ExclusiveArea this execution time is provided for.
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this execution time is described.
hardwareConfiguration	HardwareConfiguration	1	aggr	Provides information on the HardwareConfiguration used to specify this ExecutionTime.
hwElement	HwElement	0..1	ref	The hardware element (e.g. type of ECU) for which the execution time is specified.
includedLibrary	DependencyOnArtifact	*	ref	If this dependency is specified, the execution time of the library code is included in the execution time data for the runnable.
memorySectionLocation	MemorySectionLocation	*	aggr	Provides information on the MemorySectionLocation which is involved in the ExecutionTime description.
softwareContext	SoftwareContext	1	aggr	Provides information on the detailed SoftwareContext used to provide the ExecutionTime description.

Table E.50: ExecutionTime

Class	<i>FibexElement</i> (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore			
Note	ASAM FIBEX elements specifying Communication and Topology.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Subclasses	<i>BusMirrorChannelMapping</i> , <i>CommunicationCluster</i> , <i>ConsumedProvidedServiceInstanceGroup</i> , <i>CouplingElement</i> , <i>DltMessageCollectionSet</i> , <i>EcuInstance</i> , <i>Frame</i> , <i>Gateway</i> , <i>GlobalTimeDomain</i> , <i>ISignal</i> , <i>ISignalGroup</i> , <i>ISignalPduGroup</i> , <i>NmConfig</i> , <i>Pdu</i> , <i>PdurlPduGroup</i> , <i>SecureCommunicationPropsSet</i> , <i>ServiceInstanceCollectionSet</i> , <i>SoAdRoutingGroup</i> , <i>SocketConnectionIpdulIdentifierSet</i> , <i>TpConfig</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.51: FibexElement

Class	<i>HeapUsage</i> (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	Describes the heap memory usage of a SW-Component.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	MeasuredHeapUsage, RoughEstimateHeapUsage, WorstCaseHeapUsage			
Attribute	Type	Mult.	Kind	Note
hardware Configuration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this heap usage is describing.
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this heap usage usage is given.
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this heap usage is provided for.

Table E.52: HeapUsage

Class	HwElement			
Package	M2::AUTOSARTemplates::EcuResourceTemplate			
Note	This represents the ability to describe Hardware Elements on an instance level. The particular types of hardware are distinguished by the category. This category determines the applicable attributes. The possible categories and attributes are defined in HwCategory. Tags: atp.recommendedPackage=HwElements			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>HwDescriptionEntity</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
hwElement Connection	HwElementConnector	*	aggr	This represents one particular connection between two hardware elements. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime xml.sequenceOffset=110





Class	HwElement			
hwPinGroup	HwPinGroup	*	aggr	This aggregation is used to describe the connection facilities of a hardware element. Note that hardware element has no pins but only pingroups. Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime xml.sequenceOffset=90
nestedElement	HwElement	*	ref	This association is used to establish hierarchies of hw elements. Note that one particular HwElement can be target of this association only once. I.e. multiple instantiation of the same HwElement is not supported (at any hierarchy level). Stereotypes: atpVariation Tags: vh.latestBindingTime=systemDesignTime xml.sequenceOffset=70

Table E.53: HwElement

Class	HwPinGroup			
Package	M2::AUTOSARTemplates::EcuResourceTemplate			
Note	This meta-class represents the ability to describe groups of pins which are used to connect hardware elements. This group acts as a bundle of pins. Thereby they allow to describe high level connections. Pin groups can even be nested.			
Base	<i>ARObject</i> , <i>HwDescriptionEntity</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Attribute	Type	Mult.	Kind	Note
hwPinGroupContent	HwPinGroupContent	1	aggr	This aggregation describes the contained pins/pin groups.

Table E.54: HwPinGroup

Class	Identifiable (abstract)
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
Base	<i>ARObject</i> , <i>MultilanguageReferrable</i> , Referrable
Subclasses	ARPackage , AbstractEvent , AbstractImplementationDataTypeElement , AbstractServiceInstance , ApplicationEndpoint , ApplicationError , ApplicationPartitionToEcuPartitionMapping , AsynchronousServerCallResultPoint , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpFeature , AutosarOperationArgumentInstance , AutosarVariableInstance , BswInternalTriggeringPoint , BswModuleDependency , BuildActionEntity , BuildActionEnvironment , CanTpAddress , CanTpChannel , CanTpNode , Chapter , ClassContentConditional , ClientIdDefinition , ClientServerOperation , Code , CollectableElement , ComManagementMapping , CommConnectorPort , CommunicationConnector , CommunicationController , Compiler , ConsistencyNeeds , ConsumedEventGroup , CouplingPort , CouplingPortStructuralElement , CryptoServiceMapping , DataPrototypeGroup , DataTransformation , DependencyOnArtifact , DiagEventDebounceAlgorithm , DiagnosticConnectedIndicator , DiagnosticDataElement , DiagnosticFunctionInhibitSource , DiagnosticMasterToSlaveEventMapping , DiagnosticRoutineSubfunction , DltArgument , DltLogChannel , DltMessage , DolpInterface , DolpLogicAddress , ECUMapping , EOCExecutableEntityRefAbstract , EcuPartition , EcucContainerValue , EcucDefinitionElement , EcucDestinationUriDef , EcucEnumerationLiteralDef , EcucQuery , EcucValidationCondition , EndToEndProtection , ExclusiveArea ,





Class	Identifiable (abstract)			
	<p style="text-align: center;">△</p> <p><i>ExecutableEntity</i>, <i>ExecutionTime</i>, <i>FMAttributeDef</i>, <i>FMFeatureMapAssertion</i>, <i>FMFeatureMapCondition</i>, <i>FMFeatureMapElement</i>, <i>FMFeatureRelation</i>, <i>FMFeatureRestriction</i>, <i>FMFeatureSelection</i>, <i>FlatInstanceDescriptor</i>, <i>FlexrayArTpNode</i>, <i>FlexrayTpConnectionControl</i>, <i>FlexrayTpNode</i>, <i>FlexrayTpPduPool</i>, <i>FrameTriggering</i>, <i>GeneralParameter</i>, <i>GlobalTimeGateway</i>, <i>GlobalTimeMaster</i>, <i>GlobalTimeSlave</i>, <i>HeapUsage</i>, <i>HwAttributeDef</i>, <i>HwAttributeLiteralDef</i>, <i>HwPin</i>, <i>HwPinGroup</i>, <i>IPSecRule</i>, <i>IPv6ExtHeaderFilterList</i>, <i>ISignalToIPduMapping</i>, <i>ISignalTriggering</i>, <i>IdentCaption</i>, <i>InternalTriggeringPoint</i>, <i>J1939SharedAddressCluster</i>, <i>J1939TpNode</i>, <i>Keyword</i>, <i>LifeCycleState</i>, <i>LinScheduleTable</i>, <i>LinTpNode</i>, <i>Linker</i>, <i>MacMulticastGroup</i>, <i>McDataInstance</i>, <i>MemorySection</i>, <i>ModeDeclaration</i>, <i>ModeDeclarationMapping</i>, <i>ModeSwitchPoint</i>, <i>NetworkEndpoint</i>, <i>NmCluster</i>, <i>NmEcu</i>, <i>NmNode</i>, <i>NvBlockDescriptor</i>, <i>PackageableElement</i>, <i>ParameterAccess</i>, <i>PduToFrameMapping</i>, <i>PduTriggering</i>, <i>PerInstanceMemory</i>, <i>PhysicalChannel</i>, <i>PortGroup</i>, <i>PortInterfaceMapping</i>, <i>PossibleErrorReaction</i>, <i>ResourceConsumption</i>, <i>RootSwCompositionPrototype</i>, <i>RptComponent</i>, <i>RptContainer</i>, <i>RptExecutableEntity</i>, <i>RptExecutableEntityEvent</i>, <i>RptExecutionContext</i>, <i>RptProfile</i>, <i>RptServicePoint</i>, <i>RunnableEntityGroup</i>, <i>SdgAttribute</i>, <i>SdgClass</i>, <i>SecureCommunicationAuthenticationProps</i>, <i>SecureCommunicationFreshnessProps</i>, <i>ServerCallPoint</i>, <i>ServiceNeeds</i>, <i>SignalServiceTranslationElementProps</i>, <i>SignalServiceTranslationEventProps</i>, <i>SignalServiceTranslationProp</i>, <i>SocketAddress</i>, <i>SomeipTpChannel</i>, <i>SpecElementReference</i>, <i>StackUsage</i>, <i>StaticSocketConnection</i>, <i>StructuredReq</i>, <i>SwGenericAxisParamType</i>, <i>SwServiceArg</i>, <i>SwcServiceDependency</i>, <i>SwcToApplicationPartitionMapping</i>, <i>SwcToEcuMapping</i>, <i>SwcToImpMapping</i>, <i>SystemMapping</i>, <i>TcpOptionFilterList</i>, <i>TimingCondition</i>, <i>TimingConstraint</i>, <i>TimingDescription</i>, <i>TimingExtensionResource</i>, <i>TimingModelInstance</i>, <i>TlsCryptoCipherSuite</i>, <i>Topic1</i>, <i>TpAddress</i>, <i>TraceableTable</i>, <i>TraceableText</i>, <i>TracedFailure</i>, <i>TransformationProps</i>, <i>TransformationTechnology</i>, <i>Trigger</i>, <i>VariableAccess</i>, <i>VariationPointProxy</i>, <i>ViewMap</i>, <i>VlanConfig</i>, <i>WaitPoint</i></p>			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. Tags: xml.sequenceOffset=-40
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. Tags: xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. Tags: xml.sequenceOffset=-50
desc	MultiLanguageOverviewParagraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". Tags: xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. Tags: xml.sequenceOffset=-30
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR





Class	Identifiable (abstract)			
				<p>△</p> <p>models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp.</p> <p>Tags:xml.attribute=true</p>

Table E.55: Identifiable

Class	Implementation (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Description of an implementation a single software component or module.			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	BswImplementation, SwcImplementation			
Attribute	Type	Mult.	Kind	Note
buildActionManifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. Stereotypes: atpVariation Tags: vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	1..*	aggr	Specifies the provided implementation code.
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released
generatedArtifact	DependencyOnArtifact	*	aggr	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. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
hwElement	HwElement	*	ref	The hardware elements (e.g. the processor) required for this implementation.
linker	Linker	*	aggr	Specifies the linker for which this implementation has been released.
mcSupport	McSupportData	0..1	aggr	The measurement & 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. Stereotypes: atpSplitable Tags: atp.Splitkey=mcSupport
programmingLanguage	ProgramminglanguageEnum	1	attr	Programming language the implementation was created in.





Class	Implementation (abstract)			
requiredArtifact	DependencyOnArtifact	*	aggr	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. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
requiredGeneratorTool	DependencyOnArtifact	*	aggr	Relates this Implementation to a generator tool in order to generate additional artifacts during integration. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
resourceConsumption	ResourceConsumption	1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class. Stereotypes: atpSplittable Tags: atp.Splitkey=shortName
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 Bsw Implementation or for both.
swVersion	RevisionLabelString	1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.
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 E.56: Implementation

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 , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
isStructWithOptionalElement	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.





Class	ImplementationDataType			
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. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the ImplementationDataType. Stereotypes: atpSplittable Tags: 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 E.57: ImplementationDataType

Class	ImplementationDataTypeElement			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	<p>Declares a data object which is locally aggregated. Such an element can only be used within the scope where it is aggregated.</p> <p>This element either consists of further subElements or it is further defined via its swDataDefProps.</p> <p>There are several use cases within the system of ImplementationDataTypes for such a local declaration:</p> <ul style="list-style-type: none"> • It can represent the elements of an array, defining the element type and array size • It can represent an element of a struct, defining its type • It can be the local declaration of a debug element. 			
Base	<i>ARObject, AbstractImplementationDataTypeElement, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable</i>			
Attribute	Type	Mult.	Kind	Note
arraySize	PositiveInteger	0..1	attr	The existence of this attributes (if bigger than 0) defines the size of an array and declares that this ImplementationDataTypeElement represents the type of each single array element. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
arraySizeHandling	ArraySizeHandlingEnum	0..1	attr	The way how the size of the array is handled in case of a variable size array.
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls the meaning of the value of the array size.
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing ImplementationDataTypeElement as optional. This means that, at runtime, the ImplementationDataTypeElement may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the CppImplementationDataTypeElement as not valid at the sending end of a communication and determine its validity at the receiving end.





Class	ImplementationDataTypeElement			
subElement (ordered)	ImplementationDataTypeElement	*	aggr	<p>Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs").</p> <p>The aggregation of ImplementationDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime</p>
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement.

Table E.58: ImplementationDataTypeElement

Class	ImplementationProps (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Defines a symbol to be used as (depending on the concrete case) either a complete replacement or a prefix when generating code artifacts.			
Base	ARObject , Referrable			
Subclasses	BswSchedulerNamePrefix, ExecutableEntityActivationReason, SectionNamePrefix, SymbolProps , SymbolicNameProps			
Attribute	Type	Mult.	Kind	Note
symbol	CIdentifier	1	attr	The symbol to be used as (depending on the concrete case) either a complete replacement or a prefix.

Table E.59: ImplementationProps

Class	InternalBehavior (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
Base	ARObject , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	BswInternalBehavior , SwcInternalBehavior			
Attribute	Type	Mult.	Kind	Note
constant Memory	ParameterDataPrototype	*	aggr	<p>Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior.</p> <p>The shortName of ParameterDataPrototype has to be equal to the "C" identifier of the described constant.</p> <p>The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType.</p> <p>The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p>





Class	InternalBehavior (abstract)			
				<p style="text-align: center;">△</p> Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstanSpecificationMapping to be applied for the particular InternalBehavior Stereotypes: atpSplitable Tags: atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior Stereotypes: atpSplitable Tags: atp.Splitkey=dataTypeMapping
exclusiveArea	ExclusiveArea	*	aggr	This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurerment variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE. The shortName of the VariableDataPrototype has to be equal with the "C" identifier of the described variable. The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table E.60: InternalBehavior

Class	MemorySection			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::MemorySectionUsage			
Note	<p>Provides a description of an abstract memory section used in the Implementation for code or data. It shall be declared by the Implementation Description of the module or component, which actually allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the generated Implementation Description of the RTE shall contain the corresponding MemorySections.</p> <p>The attribute "symbol" (if symbol is missing: "shortName") defines the module or component specific section name used in the code. For details see the document "Specification of Memory Mapping". Typically the section name is build according the pattern:</p> <p><SwAddrMethod shortName>[_<further specialization nominator>][_<alignment>] where</p> <ul style="list-style-type: none"> • [<SwAddrMethod shortName>] is the shortName of the referenced SwAddrMethod • [<further specialization nominator>] is an optional infix to indicate the specialization in the case that several MemorySections for different purpose of the same Implementation Description referring to the same or equally named SwAddrMethods. • [<alignment>] is the alignment attributes value and is only applicable in the case that the memoryAllocationKeywordPolicy value of the referenced SwAddrMethod is set to addrMethod ShortNameAndAlignment <p>MemorySection used to Implement the code of RunnableEntitys and BswSchedulableEntitys shall have a symbol (if missing: shortName) identical to the referred SwAddrMethod to conform to the generated RTE header files.</p> <p>In addition to the section name described above, a prefix is used in the corresponding macro code in order to define a name space. This prefix is by default given by the shortName of the BswModule Description resp. the SwComponentType. It can be superseded by the prefix attribute.</p>			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
alignment	AlignmentType	0..1	attr	The attribute describes the alignment of objects within this memory section.
executableEntity	ExecutableEntity	*	ref	Reference to the ExecutableEntitites located in this section. This allows to locate different Executable Entities in different sections even if the associated Sw Addrmethod is the same. This is applicable to code sections only.
memClass Symbol	CIdentifier	0..1	attr	Defines a specific symbol in order to generate the compiler abstraction "memclass" code for this Memory Section. The existence of this attribute supersedes the usage of swAddrmethod.shortName for this purpose. The complete name of the "memclass" preprocessor symbol is constructed as <prefix>_<memClassSymbol> where prefix is defined in the same way as for the enclosing MemorySection. See also AUTOSAR_SWS_CompilerAbstraction SWS_COMPILER_00040.
option	Identifier	*	attr	This attribute introduces the ability to specify further intended properties of this MemorySection. The following two values are standardized (to be used for code sections only and exclusively to each other): <ul style="list-style-type: none"> • INLINE - The code section is declared with the compiler abstraction macro INLINE. • LOCAL_INLINE - The code section is declared with the compiler abstraction macro LOCAL_INLINE





Class	MemorySection			
				In both cases (INLINE and LOCAL_INLINE) the inline expansion depends on the compiler specific implementation of these macros. Depending on this, the code section either corresponds to an actual section in memory or is put into the section of the caller. See AUTOSAR_SWS_CompilerAbstraction for more details.
prefix	SectionNamePrefix	0..1	ref	The prefix used to set the memory section's namespace in the code. The existence of a prefix element supersedes rules for a default prefix (such as the Bsw ModuleDescription's shortName). This allows the user to define several name spaces for memory sections within the scope of one module, cluster or SWC.
size	PositiveInteger	0..1	attr	The size in bytes of the section.
swAddrmethod	SwAddrMethod	1	ref	<p>This association indicates that this module specific (abstract) memory section is part of an overall SwAddr Method, referred by the upstream declarations (e.g. calibration parameters, data element prototypes, code entities) which share a common addressing strategy. This can be evaluated for the ECU configuration of the build support.</p> <p>This association shall always be declared by the Implementation description of the module or component, which allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the software components only declare the grouping of its data prototypes to SwAddrMethods, and the generated Implementation Description of the RTE actually sets up this association.</p>
symbol	Identifier	0..1	attr	Defines the section name as explained in the main description. By using this attribute for code generation (instead of the shortName) it is possible to define several different MemorySections having the same name - e.g. symbol = CODE - but using different sectionName Prefixes.

Table E.61: MemorySection

Class	MetaDatum			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents a single meta-data item.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
length	PositiveInteger	1	attr	This attribute determines the length of the MetaDatum at run-time.
metaDatumType	TextValueSpecification	1	aggr	This aggregation contributes the specification of the concrete meta-data item type.

Table E.62: MetaDatum

Class	MetaDataItemSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the ability to define a set of meta-data items to be used in SenderReceiver Interfaces.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	1..*	ref	This reference identifies the dataElement for which the ordered list of meta-data items is defined.
metaDataItem (ordered)	MetaDataItem	*	aggr	This aggregation represents the ordered definition of meta-data items.

Table E.63: MetaDataItemSet

Class	ModeDeclaration			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model. Tags: atp.ManifestKind=ExecutionManifest,MachineManifest			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
value	PositiveInteger	0..1	attr	The RTE shall take the value of this attribute for generating the source code representation of this Mode Declaration.

Table E.64: ModeDeclaration

Class	ModeDeclarationGroup			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. Tags: atp.ManifestKind=ExecutionManifest,MachineManifest atp.recommendedPackage=ModeDeclarationGroups			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
initialMode	ModeDeclaration	1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
mode Declaration	ModeDeclaration	1..*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. Stereotypes: atpVariation Tags: vh.latestBindingTime=blueprintDerivationTime
modeManager ErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user).
modeTransition	ModeTransition	*	aggr	This represents the available ModeTransitions of the ModeDeclarationGroup
modeUserError Behavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager).





Class	ModeDeclarationGroup			
onTransition Value	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses.

Table E.65: ModeDeclarationGroup

Class	ModeDeclarationGroupPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context. Tags: atp.ManifestKind=ExecutionManifest,MachineManifest			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mult.	Kind	Note
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time.
type	ModeDeclarationGroup	1	tref	The "collection of ModeDeclarations" (= ModeDeclaration Group) supported by a component Stereotypes: isOfType

Table E.66: ModeDeclarationGroupPrototype

Class	NonqueuedReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to non-queued receiving.			
Base	ARObject, RPortComSpec, ReceiverComSpec			
Attribute	Type	Mult.	Kind	Note
aliveTimeout	TimeValue	1	attr	Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description. If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.
enableUpdate	Boolean	1	attr	This attribute controls whether application code is entitled to check whether the value of the corresponding VariableDataPrototype has been updated.
filter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement.
handleData Status	Boolean	0..1	attr	If this attribute is set to true than the Rte_IStatus API shall exist. If the attribute does not exist or is set to false then the Rte_IStatus API may still exist in response to the existence of further conditions.
handleNever Received	Boolean	1	attr	This attribute specifies whether for the corresponding VariableDataPrototype the "never received" flag is available. If yes, the RTE is supposed to assume that initially the VariableDataPrototype has not been received before. After the first reception of the corresponding VariableDataPrototype the flag is cleared.





Class		NonqueuedReceiverComSpec		
				<ul style="list-style-type: none"> If the value of this attribute is set to "true" the flag is required. If set to "false", the RTE shall not support the "never received" functionality for the corresponding VariableDataPrototype.
handleTimeout Type	HandleTimeoutEnum	1	attr	This attribute controls the behavior with respect to the handling of timeouts.
initValue	ValueSpecification	0..1	aggr	Initial value to be used in case the sending component is not yet initialized. If the sender also specifies an initial value the receiver's value will be used.
timeout Substitution Value	ValueSpecification	0..1	aggr	This attribute represents the substitution value applicable in the case of a timeout.

Table E.67: NonqueuedReceiverComSpec

Class		NumericalValueSpecification		
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	A numerical ValueSpecification which is intended to be assigned to a Primitive data element. Note that the numerical value is a variant, it can be computed by a formula.			
Base	ARObject , ValueSpecification			
Attribute	Type	Mult.	Kind	Note
value	Numerical	1	attr	This is the value itself. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.68: NumericalValueSpecification

Class		PPortPrototype		
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port providing a certain port interface.			
Base	ARObject , AbstractProvidedPortPrototype , AtpBlueprintable , AtpFeature , AtpPrototype , Identifiable , MultilanguageReferrable , PortPrototype , Referrable			
Attribute	Type	Mult.	Kind	Note
provided Interface	PortInterface	1	tref	The interface that this port provides. Stereotypes: isOfType

Table E.69: PPortPrototype

Class		PRPortPrototype		
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	ARObject , AbstractProvidedPortPrototype , AbstractRequiredPortPrototype , AtpBlueprintable , AtpFeature , AtpPrototype , Identifiable , MultilanguageReferrable , PortPrototype , Referrable			





Class	PRPortPrototype			
Attribute	Type	Mult.	Kind	Note
provided Required Interface	PortInterface	1	tref	This represents the PortInterface used to type the PRPort Prototype Stereotypes: isOfType

Table E.70: PRPortPrototype

Class	PackageableElement (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	This meta-class specifies the ability to be a member of an AUTOSAR package.			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>ARElement</i> , EnumerationMappingTable, <i>FibexElement</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.71: PackageableElement

Class	ParameterDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A parameter element used for parameter interface and internal behavior, supporting signal like parameter and characteristic value communication patterns and parameter and characteristic value definition.			
Base	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>AutosarDataPrototype</i> , <i>DataPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype

Table E.72: ParameterDataPrototype

Class	ParameterInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A parameter interface declares a number of parameter and characteristic values to be exchanged between parameter components and software components. Tags: atp.recommendedPackage=PortInterfaces			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>DataInterface</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>PortInterface</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
parameter	ParameterDataPrototype	1..*	aggr	The ParameterDataPrototype of this ParameterInterface.

Table E.73: ParameterInterface

Class	ParameterSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The ParameterSwComponentType defines parameters and characteristic values accessible via provided Ports. The provided values are the same for all connected SwComponentPrototypes Tags: atp.recommendedPackage=SwComponentTypes			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Attribute	Type	Mult.	Kind	Note
constant Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstanSpecificationMapping to be applied for the particular ParameterSwComponentType Stereotypes: atpSplitable Tags: atp.Splitkey=constantMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular ParameterSwComponentType Stereotypes: atpSplitable Tags: atp.Splitkey=dataTypeMapping
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	The purpose of this is that within the context of a given SwComponentType some data def properties of individual instantiations can be modified. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of PortPrototypes Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.74: ParameterSwComponentType

Class	PassThroughSwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	This kind of SwConnector can be used inside a CompositionSwComponentType to connect two delegation PortPrototypes.			
Base	ARObject , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable , SwConnector			
Attribute	Type	Mult.	Kind	Note
providedOuter Port	AbstractProvidedPort Prototype	1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector.
requiredOuter Port	AbstractRequiredPort Prototype	1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector.

Table E.75: PassThroughSwConnector

Class	PhysConstrs			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express physical constraints. Therefore it has (in opposite to InternalConstrs) a reference to a Unit.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note





Class	PhysConstrs			
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the constraint. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
maxDiff	Numerical	0..1	attr	Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis. Tags: xml.sequenceOffset=60
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in curves and maps. Tags: xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This specifies the monotony constraints on the data object. Note that this applies only to curves and maps. Tags: xml.sequenceOffset=70
scaleConstr (ordered)	ScaleConstr	*	aggr	This is one particular scale which contributes to the data constraints. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false
unit	Unit	0..1	ref	This is the unit to which the physical constraints relate to. In particular, it is the physical unit of the specified limits. Tags: xml.sequenceOffset=80
upperLimit	Limit	0..1	attr	This specifies the upper limit of the constraint. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table E.76: PhysConstrs

Class	PncMappingIdent			
Package	M2::AUTOSARTemplates::SystemTemplate::PncMapping			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable PncMapping.			
Base	ARObject, Referrable			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

Table E.77: PncMappingIdent

Class	PortAPIOption
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions
Note	Options how to generate the signatures of calls for an AtomicSwComponentType in order to communicate over a PortPrototype (for calls into a RunnableEntity as well as for calls from a RunnableEntity to the PortPrototype).





Class		PortAPIOption		
Base		<i>AObject</i>		
Attribute	Type	Mult.	Kind	Note
enableTakeAddress	Boolean	1	attr	If set to true, the software-component is able to use the API reference for deriving a pointer to an object.
errorHandling	DataTransformationErrorHandlingEnum	0..1	attr	This specifies whether a RunnableEntity accessing a Port Prototype that is referenced by this PortAPIOption shall specifically handle transformer errors or not.
indirectAPI	Boolean	1	attr	If set to true this attribute specifies an "indirect API" to be generated for the associated port which means that the software-component is able to access the actions on a port via a pointer to an object representing a port. This allows e.g. iterating over ports in a loop. This option has no effect for PPortPrototypes of client/server interfaces.
port	PortPrototype	1	ref	The option is valid for generated functions related to communication over this port
portArgValue (ordered)	PortDefinedArgumentValue	*	aggr	An argument value defined by this port.
supportedFeature	SwcSupportedFeature	*	aggr	This collection specifies which features are supported by the RunnableEntities which access a PortPrototype that it referenced by this PortAPIOption.
transformerStatusForwarding	DataTransformationStatusForwardingEnum	0..1	attr	This specifies whether a RunnableEntity accessing a Port Prototype that is referenced by this PortAPIOption shall be able to forward a status to the transformer chain. Tags: atp.Status=draft

Table E.78: PortAPIOption

Class		PortGroup		
Package		M2::AUTOSARTemplates::SWComponentTemplate::Components		
Note		Group of ports which share a common functionality, e.g. need specific network resources. This information shall be available on the VFB level in order to delegate it properly via compositions. When propagated into the ECU extract, this information is used as input for the configuration of Services like the Communication Manager. A PortGroup is defined locally in a component (which can be a composition) and refers to the "outer" ports belonging to the group as well as to the "inner" groups which propagate this group into the components which are part of a composition. A PortGroup within an atomic SWC cannot be linked to inner groups.		
Base		<i>AObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , Identifiable , MultilanguageReferrable , Referrable		
Attribute	Type	Mult.	Kind	Note
innerGroup	PortGroup	*	iref	Links a PortGroup in a composition to another PortGroup, that is defined in a component which is part of this CompositionSwComponentType.
outerPort	PortPrototype	*	ref	Outer PortPrototype of this AtomicSwComponentType which belongs to the group. A port can belong to several groups or to no group at all. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.79: PortGroup

Class	PortInterface (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	ClientServerInterface , DataInterface , ModeSwitchInterface , TriggerInterface			
Attribute	Type	Mult.	Kind	Note
isService	Boolean	1	attr	This flag is set if the PortInterface is to be used for communication between an <ul style="list-style-type: none"> • ApplicationSwComponentType or • ServiceProxySwComponentType or • SensorActuatorSwComponentType or • ComplexDeviceDriverSwComponentType • ServiceSwComponentType • EcuAbstractionSwComponentType and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

Table E.80: PortInterface

Class	PortInterfaceMapping (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies one PortInterfaceMapping to support the connection of Ports typed by two different Port Interfaces with PortInterface elements having unequal names and/or unequal semantic (resolution or range).			
Base	ARObject , AtpBlueprint , AtpBlueprintable , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	ClientServerInterfaceMapping , ModelInterfaceMapping , TriggerInterfaceMapping , VariableAndParameterInterfaceMapping			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.81: PortInterfaceMapping

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			
Attribute	Type	Mult.	Kind	Note
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.





Class	PortPrototype (abstract)			
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServer 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 volatile 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 E.82: PortPrototype

Class	RPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port requiring a certain port interface.			
Base	<i>ARObject</i> , <i>AbstractRequiredPortPrototype</i> , <i>AtpBlueprintable</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PortPrototype</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
required Interface	PortInterface	1	tref	The interface that this port requires. Stereotypes: isOfType

Table E.83: RPortPrototype

Class	RTEEvent (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	Abstract base class for all RTE-related events			
Base	<i>ARObject</i> , <i>AbstractEvent</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, TimingEvent, TransformerHardErrorEvent			
Attribute	Type	Mult.	Kind	Note
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. Stereotypes: atpSplittable Tags: atp.Splitkey=contextPort, contextModeDeclaration GroupPrototype, targetModeDeclaration
startOnEvent	RunnableEntity	0..1	ref	RunnableEntity starts when the corresponding RTEEvent occurs.

Table E.84: RTEEvent

Class	<i>ReceiverComSpec</i> (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	<i>ARObject</i> , <i>RPortComSpec</i>			
Subclasses	NonqueuedReceiverComSpec , QueuedReceiverComSpec			
Attribute	Type	Mult.	Kind	Note
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a ReceiverComSpec. The purpose of this aggregation is to be able to specify the network representation of leaf elements of Application CompositeDataTypes.
dataElement	AutosarDataPrototype	0..1	ref	Data element these attributes belong to.
dataUpdate Period	TimeValue	0..1	attr	This attribute defines the period in which the application shall check for updated data. This attribute is used for the configuration of the E2E protection. Tags: atp.Status=draft
handleOutOf Range	HandleOutOfRange Enum	1	attr	This attribute controls how values that are out of the specified range are handled according to the values of HandleOutOfRangeEnum.
handleOutOf RangeStatus	HandleOutOfRange StatusEnum	0..1	attr	Control the way how return values are created in case of an out-of-range situation.
maxDelta CounterInit	PositiveInteger	0..1	attr	Initial maximum allowed gap between two counter values of two consecutively received valid Data, i.e. how many subsequent lost data is accepted. For example, if the receiver gets Data with counter 1 and MaxDeltaCounter Init is 1, then at the next reception the receiver can accept Counters with values 2 and 3, but not 4. Note that if the receiver does not receive new Data at a consecutive read, then the receiver increments the tolerance by 1. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	The maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.
network Representation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus.
replaceWith	VariableAccess	0..1	aggr	This aggregation is used to identify the AutosarData Prototype to be taken for sourcing an external replacement in the out-of-range handling.
syncCounterInit	PositiveInteger	0..1	attr	Number of Data required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.
usesEndToEnd Protection	Boolean	0..1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.85: ReceiverComSpec

Class	Referrable (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	ARObject			
Subclasses	AtpDefinition , BswDistinguishedPartition , BswModuleCallPoint , BswModuleClientServerEntry , BswVariableAccess , CouplingPortTrafficClassAssignment , DiagnosticDebounceAlgorithmProps , DiagnosticEnvModeElement , EthernetPriorityRegeneration , EventHandler , ExclusiveAreaNestingOrder , HwDescriptionEntity , ImplementationProps , LinSlaveConfigIdent , ModeTransition , MultilanguageReferrable , PduActivationRoutingGroup , PncMappingIdent , SingleLanguageReferrable , SoConIPduIdentifier , SocketConnectionBundle , TimeSyncServerConfiguration , TpConnectionIdent			
Attribute	Type	Mult.	Kind	Note
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. Tags: xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. Tags: xml.sequenceOffset=-90

Table E.86: Referrable

Class	RoleBasedPortAssignment			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPort Prototype) of an AtomicSwComponentType. With this assignment, the role of the service port can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct connector.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
portPrototype	PortPrototype	1	ref	Service PortPrototype used in the assigned role. This PortPrototype shall either belong to the same AtomicSw ComponentType as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockSw ComponentType as the NvBlockDescriptor.
role	Identifier	1	attr	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a Port Interface as standardized in the Software Specification of the related AUTOSAR Service.

Table E.87: RoleBasedPortAssignment

Class	RunnableEntity			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	A RunnableEntity represents the smallest code-fragment that is provided by an AtomicSwComponent Type and are executed under control of the RTE. RunnableEntities are for instance set up to respond to data reception or operation invocation on a server.			
Base	ARObject, AtpClassifier , AtpFeature , AtpStructureElement , ExecutableEntity , Identifiable , MultilanguageReferrable , Referrable			





Class	RunnableEntity			
Attribute	Type	Mult.	Kind	Note
argument (ordered)	RunnableEntity Argument	*	aggr	This represents the formal definition of a an argument to a RunnableEntity.
asynchronous ServerCall ResultPoint	AsynchronousServer CallResultPoint	*	aggr	<p>The server call result point admits a runnable to fetch the result of an asynchronous server call.</p> <p>The aggregation of AsynchronousServerCallResultPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes and the variant existence of server call result points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
canBelInvoked Concurrently	Boolean	1	attr	If the value of this attribute is set to "true" the enclosing RunnableEntity can be invoked concurrently (even for one instance of the corresponding AtomicSwComponent Type). This implies that it is the responsibility of the implementation of the RunnableEntity to take care of this form of concurrency. Note that the default value of this attribute is set to "false".
dataRead Access	VariableAccess	*	aggr	<p>RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataReadAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataReadAccess in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataReceive PointBy Argument	VariableAccess	*	aggr	<p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of an argument in the function signature.</p> <p>The aggregation of dataReceivePointByArgument is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data receive points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataReceive PointByValue	VariableAccess	*	aggr	<p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The result is passed back to the application by means of the return value. The aggregation of dataReceivePointBy Value is subject to variability with the purpose to support the conditional existence of sender receiver ports or the</p>





Class	RunnableEntity			
				<p style="text-align: center;">△</p> variant existence of data receive points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataSendPoint	VariableAccess	*	aggr	RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataSendPoint is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data send points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataWrite Access	VariableAccess	*	aggr	RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
external TriggeringPoint	ExternalTriggeringPoint	*	aggr	The aggregation of ExternalTriggeringPoint is subject to variability with the purpose to support the conditional existence of trigger ports or the variant existence of external triggering points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=ident.shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internal TriggeringPoint	InternalTriggeringPoint	*	aggr	The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeAccess Point	ModeAccessPoint	*	aggr	The runnable has a mode access point. The aggregation of ModeAccessPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode access points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=ident.shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime





Class	RunnableEntity			
modeSwitch Point	ModeSwitchPoint	*	aggr	<p>The runnable has a mode switch point. The aggregation of ModeSwitchPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode switch points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
parameter Access	ParameterAccess	*	aggr	<p>The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a Parameter DataPrototype which may either be local or within a Port Prototype.</p> <p>The aggregation of ParameterAccess is subject to variability with the purpose to support the conditional existence of parameter ports and component local parameters as well as the variant existence of Parameter Access (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
readLocal Variable	VariableAccess	*	aggr	<p>The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of readLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of read LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
serverCallPoint	ServerCallPoint	*	aggr	<p>The RunnableEntity has a ServerCallPoint. The aggregation of ServerCallPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes or the variant existence of server call points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
symbol	CIdentifier	1	attr	<p>The symbol describing this RunnableEntity's entry point. This is considered the API of the RunnableEntity and is required during the RTE contract phase.</p>
waitPoint	WaitPoint	*	aggr	<p>The WaitPoint associated with the RunnableEntity.</p>
writtenLocal Variable	VariableAccess	*	aggr	<p>The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional</p>





Class	RunnableEntity			
				<p>existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>

Table E.88: RunnableEntity

Class	SenderComSpec (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a sender port (PPortPrototype typed by SenderReceiverInterface).			
Base	ARObject, PPortComSpec			
Subclasses	NonqueuedSenderComSpec, QueuedSenderComSpec			
Attribute	Type	Mult.	Kind	Note
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a SenderComSpec.
dataElement	AutosarDataPrototype	0..1	ref	Data element these quality of service attributes apply to.
dataUpdate Period	TimeValue	0..1	attr	This attribute describes the period in which the applications are assumed to transmit E2E-protected messages. Tags: atp.Status=draft
handleOutOf Range	HandleOutOfRange Enum	1	attr	This attribute controls how out-of-range values shall be dealt with.
network Representation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus.
transmission Acknowledge	Transmission Acknowledgement Request	0..1	aggr	Requested transmission acknowledgement for data element.
usesEndToEnd Protection	Boolean	0..1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime

Table E.89: SenderComSpec

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			
Attribute	Type	Mult.	Kind	Note





Class	SenderReceiverInterface			
dataElement	VariableDataPrototype	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement
metaDataItemSet	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with dataElements of the enclosing SenderReceiverInterface

Table E.90: SenderReceiverInterface

Class	SensorActuatorSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The SensorActuatorSwComponentType introduces the possibility to link from the software representation of a sensor/actuator to its hardware description provided by the ECU Resource Template. Tags: atp.recommendedPackage=SwComponentTypes			
Base	ARElement , ARObject , AtomicSwComponentType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Attribute	Type	Mult.	Kind	Note
sensorActuator	HwDescriptionEntity	1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware.

Table E.91: SensorActuatorSwComponentType

Class	ServiceNeeds (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This expresses the abstract needs that a Software Component or Basic Software Module has on the configuration of an AUTOSAR Service to which it will be connected. "Abstract needs" means that the model abstracts from the Configuration Parameters of the underlying Basic Software.			
Base	ARObject , Identifiable , MultilanguageReferrable , Referrable			
Subclasses	BswMgrNeeds, ComMgrUserNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, <i>DiagnosticCapabilityElement</i> , DltUserNeeds, <i>DolpServiceNeeds</i> , EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, GlobalSupervisionNeeds, HardwareTestNeeds, IndicatorStatusNeeds, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequestServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpointNeeds, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, V2xFacUserNeeds, V2xMUserNeeds, VendorSpecificServiceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.92: ServiceNeeds

Class	ServiceProxySwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This class provides the ability to express a software-component which provides access to an internal service for remote ECUs. It acts as a proxy for the service providing access to the service. An important use case is the request of vehicle mode switches: Such requests can be communicated via sender-receiver interfaces across ECU boundaries, but the mode manager being responsible to perform			





Class	ServiceProxySwComponentType			
	<p>the mode switches is an AUTOSAR Service which is located in the Basic Software and is not visible in the VFB view. To handle this situation, a ServiceProxySwComponentType will act as proxy for the mode manager. It will have R-Ports to be connected with the mode requestors on VFB level and Service-Ports to be connected with the local mode manager at ECU integration time.</p> <p>Apart from the semantics, a ServiceProxySwComponentType has these specific properties:</p> <ul style="list-style-type: none"> • A prototype of it can be mapped to more than one ECUs in the system description. • Exactly one additional instance of it will be created in the ECU-Extract per ECU to which the prototype has been mapped. • For remote communication, it can have only R-Ports with sender-receiver interfaces and 1:n semantics. • There shall be no connectors between two prototypes of any ServiceProxySwComponentType. <p>Tags:atp.recommendedPackage=SwComponentTypes</p>			
Base	ARElement , ARObject , AtomicSwComponentType , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , SwComponentType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.93: ServiceProxySwComponentType

Enumeration	ServiceVersionAcceptanceKindEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	<p>Defined the possible acceptance kinds for required service instances.</p> <p>Tags:atp.Status=draft</p>
Literal	Description
exactOrAnyMinorVersion	<p>Search for ANY or specific minor version service instance and select either ALL returned service instances (in case of ANY) or exactly the specific minor version service instances defined in required MinorVersion.</p> <p>Tags:atp.EnumerationLiteralIndex=0</p>
minimumMinorVersion	<p>Search for ANY minor version service instance and select only those service instances which have an equal or greater minor version than given in requiredMinorVersion.</p> <p>Tags:atp.EnumerationLiteralIndex=1</p>

Table E.94: ServiceVersionAcceptanceKindEnum

Class	SignalPathConstraint (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	Additional guidelines for the System Generator, which specific way a signal between two Software Components should take in the network without defining in which frame and with which timing it is transmitted.			
Base	ARObject			
Subclasses	CommonSignalPath , ForbiddenSignalPath , PermissibleSignalPath , SeparateSignalPath			
Attribute	Type	Mult.	Kind	Note





Class	SignalPathConstraint (abstract)			
introduction	DocumentationBlock	1	aggr	This represents introductory documentation about the signal path constraint.

Table E.95: SignalPathConstraint

Class	SoAdRoutingGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ObsoleteModel			
Note	Routing of Pdus in the SoAd can be activated or deactivated. The ShortName of this element shall contain the RoutingGroupId. Tags: atp.Status=obsolete atp.recommendedPackage=SoAdRoutingGroups			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
eventGroupControlType	EventGroupControlTypeEnum	0..1	attr	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.

Table E.96: SoAdRoutingGroup

Class	SocketConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ObsoleteModel			
Note	The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack. Tags: atp.Status=obsolete			
Base	<i>ARObject</i> , <i>Describable</i>			
Attribute	Type	Mult.	Kind	Note
clientIpAddrFromConnectionRequest	Boolean	0..1	attr	If set to true the Server "learns" the client IP address on connection request. This means that the statically configured IP Address of the related client shall be ignored. If set to false the Server only accepts statically configured IP address, e.g. 192.168.1.2. This means that the statically configured IP Address of the Client shall be used.
clientPort	SocketAddress	0..1	ref	Client Port for TCP/UDP connection in an abstract communication sense. The client is the major requester of the communication. Please note that the client may also produce data. Tags: atp.Status=obsolete





Class	SocketConnection			
clientPortFromConnectionRequest	Boolean	0..1	attr	If set to true the Server "learns" the client Port on connection request. This means that the statically configured Port of the related client shall be ignored. If set to false the Server only accepts statically configured Port. This means that the statically configured Port of the Client shall be used.
pdu	SocketConnectionIpduIdentifier	*	aggr	PDUs handed over by the PDU Router (Transmission over the Ethernet) or PDUs handed over by SoAd (Reception over Ethernet). Multiple IPdus can be transmitted over one socket connection. Tags: atp.Status=obsolete
pduCollectionMaxBufferSize	PositiveInteger	0..1	attr	Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.
pduCollectionTimeout	TimeValue	0..1	attr	Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.
runtimeIpAddressConfiguration	RuntimeAddressConfigurationEnum	0..1	attr	This attribute determines which protocol is used by the client to obtain the IP Address information. If this attribute is not set to none the value determines the service used by the client to obtain the IP Address information for the SocketConnection. If this attribute is set to none the client used the statically configured IP Address information.
runtimePortConfiguration	RuntimeAddressConfigurationEnum	0..1	attr	This attribute determines which protocol is used by the client to obtain the Port information. If this attribute is not set to none the value determines the service used by the client to obtain the Port information for the Socket Connection. If this attribute is set to none the client uses the statically configured Port information.
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the SocketConnection. It shall be unique within its context.

Table E.97: SocketConnection

Class	SocketConnectionBundle			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ObsoleteModel			
Note	This elements groups SocketConnections, i.e. specifies socket connections belonging to the bundle and describes properties which are common for all socket connections in the bundle. Tags: atp.Status=obsolete			
Base	ARObject, <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
bundledConnection	SocketConnection	1..*	aggr	Collection of SocketConnections in the connectionGroup. Tags: atp.Status=obsolete
differentiatedServiceField	PositiveInteger	0..1	attr	The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.





Class	SocketConnectionBundle			
flowLabel	PositiveInteger	0..1	attr	The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.
pathMtuDiscoveryEnabled	Boolean	0..1	attr	Defines whether the Path MTU Discovery shall be performed for the related socket.
pdu	SocketConnectionIpduIdentifier	*	aggr	With this aggregation SocketConnectionIpduIdentifier elements are assigned to all SocketConnections that are available in this SocketConnectionBundle. Tags: atp.Status=obsolete
udpChecksumHandling	UdpChecksumCalculationEnum	0..1	attr	Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksumDisabled) on the related socket connection.

Table E.98: SocketConnectionBundle

Class	SomeipSdClientServiceInstanceConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Client specific settings that are relevant for the configuration of SOME/IP Service-Discovery. Tags: atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=SomeipSdTimingConfigs			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
initialFindBehavior	InitialSdDelayConfig	0..1	aggr	Controls initial find behavior of clients.
serviceFindTimeToLive	PositiveInteger	1	attr	This attribute represents the ability to define the time in seconds the service find is valid.

Table E.99: SomeipSdClientServiceInstanceConfig

Class	SomeipServiceVersion			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This meta-class represents the ability to describe a version of a SOME/IP Service. Tags: atp.ManifestKind=ServiceInstanceManifest atp.Status=draft			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. Tags: xml.sequenceOffset=10
minorVersion	PositiveInteger	1	attr	Minor Version of the ServiceInterface. Tags: xml.sequenceOffset=20

Table E.100: SomeipServiceVersion

Class	StackUsage (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	Describes the stack memory usage of a software.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	MeasuredStackUsage, RoughEstimateStackUsage, WorstCaseStackUsage			
Attribute	Type	Mult.	Kind	Note
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this stack usage is described.
hardware Configuration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this stack usage is describing.
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this stack usage is given.
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this stack usage is provided for.

Table E.101: StackUsage

Primitive	String
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	<p>This represents a String in which white-space shall be normalized before processing. For example: in order to compare two Strings:</p> <ul style="list-style-type: none"> • leading and trailing white-space needs to be removed • consecutive white-space (blank, cr, lf, tab) needs to be replaced by one blank. <p>Tags: xml.xsd.customType=STRING xml.xsd.type=string</p>

Table E.102: String

Class	SwBaseType			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	<p>This meta-class represents a base type used within ECU software.</p> <p>Tags:atp.recommendedPackage=BaseTypes</p>			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , BaseType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.103: SwBaseType

Class	SwComponentPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	ARObject, AtpFeature , AtpPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note





Class	SwComponentPrototype			
type	SwComponentType	1	tref	Type of the instance. Stereotypes: isOfType

Table E.104: SwComponentPrototype

Class	SwComponentType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for AUTOSAR software components.			
Base	ARElement , ARObject , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Subclasses	AtomicSwComponentType , CompositionSwComponentType , ParameterSwComponentType			
Attribute	Type	Mult.	Kind	Note
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
port	PortPrototype	*	aggr	The PortPrototypes through which this SwComponent Type can communicate. The aggregation of PortPrototype is subject to variability with the purpose to support the conditional existence of PortPrototypes. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the SwComponentType. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=swComponentDocumentation, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which UnitGroups are relevant in the context of referencing SwComponentType.

Table E.105: SwComponentType

Class	SwConnector (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	The base class for connectors between ports. Connectors have to be identifiable to allow references from the system constraint template.			
Base	ARObject , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , Referrable			





Class	SwConnector (abstract)			
Subclasses	AssemblySwConnector , DelegationSwConnector , PassThroughSwConnector			
Attribute	Type	Mult.	Kind	Note
mapping	PortInterfaceMapping	0..1	ref	Reference to a PortInterfaceMapping specifying the mapping of unequal named PortInterface elements of the two different PortInterfaces typing the two PortPrototypes which are referenced by the ConnectorPrototype .

Table E.106: SwConnector

Class	<<atpVariation>> SwDataDefProps			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	<p>This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated.</p> <p>Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations.</p> <p>SwDataDefProps covers various aspects:</p> <ul style="list-style-type: none"> • Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet • Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, swAddrMethod, swPointerTargetProps, baseType, implementationDataType and additionalNativeTypeQualifier • Access policy for the MCD system, mainly expressed by swCalibrationAccess • Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue • Code generation policy provided by swRecordLayout <p>Tags:vh.latestBindingTime=codeGenerationTime</p>			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
additionalNativeTypeQualifier	NativeDeclarationString	0..1	attr	This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string. Tags: xml.sequenceOffset=235
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. Tags: xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. Tags: xml.sequenceOffset=50





Class	<<atpVariation>> SwDataDefProps			
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. Tags: xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. Tags: xml.sequenceOffset=190
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. Tags: xml.sequenceOffset=210
displayPresentation	DisplayPresentationEnum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.
implementationDataType	AbstractImplementationDataType	0..1	ref	This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially <ul style="list-style-type: none"> • redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype • the target type of a pointer (see SwPointerTarget Props), if it does not refer to a base type directly • the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly • the data type of an SwServiceArg, if it does not refer to a base type directly Tags: xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. Tags: xml.sequenceOffset=255
stepSize	Float	0..1	attr	This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself. Tags: xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memory AllocationKeywordPolicy of the referenced SwAddr Method. Tags: xml.sequenceOffset=33
swBitRepresentation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. Tags: xml.sequenceOffset=60
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	Specifies the read or write access by MCD tools for this data object. Tags: xml.sequenceOffset=70





Class	<<atpVariation>> SwDataDefProps			
swCalprmAxis Set	SwCalprmAxisSet	0..1	aggr	This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters. Tags: xml.sequenceOffset=90
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. Tags: xml.sequenceOffset=170 xml.typeElement=false
swData Dependency	SwDataDependency	0..1	aggr	Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system). Tags: xml.sequenceOffset=200
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. Tags: xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. Tags: xml.sequenceOffset=230
swIntended Resolution	Numerical	0..1	attr	The purpose of this element is to describe the requested quantization of data objects early on in the design process. The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula). In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution. The resolution is specified in the physical domain according to the property "unit". Tags: xml.sequenceOffset=240
swInterpolation Method	Identifier	0..1	attr	This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked. Tags: xml.sequenceOffset=250
swIsVirtual	Boolean	0..1	attr	This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency . Tags: xml.sequenceOffset=260
swPointerTarget Props	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. Tags: xml.sequenceOffset=280
swRecord Layout	SwRecordLayout	0..1	ref	Record layout for this data object. Tags: xml.sequenceOffset=290





Class	<<atpVariation>> SwDataDefProps			
swRefresh Timing	MultidimensionalTime	0..1	aggr	<p>This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system.</p> <p>So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing.</p> <p>Tags:xml.sequenceOffset=300</p>
swTextProps	SwTextProps	0..1	aggr	<p>the specific properties if the data object is a text object.</p> <p>Tags:xml.sequenceOffset=120</p>
swValueBlock Size	Numerical	0..1	attr	<p>This represents the size of a Value Block</p> <p>Stereotypes: atpVariation</p> <p>Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=80</p>
swValueBlock SizeMult (ordered)	Numerical	*	attr	<p>This attribute is used to specify the dimensions of a value block (VAL_BLK) for the case that that value block has more than one dimension.</p> <p>The dimensions given in this attribute are ordered such that the first entry represents the first dimension, the second entry represents the second dimension, and so on.</p> <p>For one-dimensional value blocks the attribute swValueBlockSize shall be used and this attribute shall not exist.</p> <p>Stereotypes: atpVariation</p> <p>Tags:vh.latestBindingTime=preCompileTime</p>
unit	Unit	0..1	ref	<p>Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible.</p> <p>Tags:xml.sequenceOffset=350</p>
valueAxisData Type	ApplicationPrimitive DataType	0..1	ref	<p>The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType.</p> <p>Tags:xml.sequenceOffset=355</p>

Table E.107: SwDataDefProps

Enumeration	SwImplPolicyEnum
Package	M2::MSR::DataDictionary::DataDefProperties
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Literal	Description
const	<p>forced implementation such that the running software within the ECU shall not modify it. For example implemented with the "const" modifier in C. This can be applied for parameters (not for those in NVRAM) as well as argument data prototypes.</p> <p>Tags:atp.EnumerationLiteralIndex=0</p>





Enumeration	SwImplPolicyEnum
fixed	This data element is fixed. In particular this indicates, that it might also be implemented e.g. as in place data, (#DEFINE). Tags: atp.EnumerationLiteralIndex=1
measurementPoint	The data element is created for measurement purposes only. The data element is never read directly within the ECU software. In contrast to a "standard" data element in an unconnected provide port is, this unconnection is guaranteed for measurementPoint data elements. Tags: atp.EnumerationLiteralIndex=2
queued	The content of the data element is queued and the data element has 'event' semantics, i.e. data elements are stored in a queue and all data elements are processed in 'first in first out' order. The queuing is intended to be implemented by RTE Generator. This value is not applicable for parameters. Tags: atp.EnumerationLiteralIndex=3
standard	This is applicable for all kinds of data elements. For variable data prototypes the 'last is best' semantics applies. For parameter there is no specific implementation directive. Tags: atp.EnumerationLiteralIndex=4

Table E.108: SwImplPolicyEnum

Class	SwRecordLayout			
Package	M2::MSR::DataDictionary::RecordLayout			
Note	Defines how the data objects (variables, calibration parameters etc.) are to be stored in the ECU memory. As an example, this definition specifies the sequence of axis points in the ECU memory. Iterations through axis values are stored within the sub-elements swRecordLayoutGroup. Tags: atp.recommendedPackage=SwRecordLayouts			
Base	ARElement , ARObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
swRecordLayoutGroup	SwRecordLayoutGroup	1	aggr	This is the top level record layout group. Tags: xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table E.109: SwRecordLayout

Class	SwRecordLayoutGroup			
Package	M2::MSR::DataDictionary::RecordLayout			
Note	Specifies how a record layout is set up. Using SwRecordLayoutGroup it recursively models iterations through axis values. The subelement swRecordLayoutGroupContentType may reference other SwRecordLayouts, SwRecordLayoutVs and SwRecordLayoutGroups for the modeled record layout.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note





Class	SwRecordLayoutGroup			
category	AsamRecordLayout Semantics	0..1	attr	<p>This attribute denotes the semantics in particular in terms of the corresponding A2L-Keyword. This is to support the mapping of the more general record layouts in AUTOSAR/MSR to the specific A2l keywords.</p> <p>It is possible to express the specific semantics of A2l recordlayout keywords in swRecordlayoutGroup but not always vice versa. Therefore the mapping is provided in this optional attribute.</p> <p>Tags:xml.sequenceOffset=5</p>
desc	MultiLanguageOverview Paragraph	0..1	aggr	<p>This aggregation allows a brief description about the particular record layout group which can help to identify the entry. In-depth documentation should be added to the introduction of the surrounding record layout.</p> <p>Tags:xml.sequenceOffset=20</p>
shortLabel	Identifier	1	attr	<p>This attribute specifies a name which can be used e.g. when ECU code is generated from the record layout group.</p> <p>Tags:xml.sequenceOffset=3</p>
swGenericAxis ParamType	SwGenericAxisParam Type	0..1	ref	<p>This association allows to specify record layout groups to iterate over generic axis parameters. For example, if the generic axis parameter is an array, the record layout group will iterate over this array.</p> <p>Obviously, the axis referred to by swRecordLayoutGroup Axis shall be a generic axis in which the referenced Sw GenericAxisType is aggregated.</p> <p>Tags:xml.sequenceOffset=50</p>
swRecord Layout Component	Identifier	0..1	attr	<p>This attribute is used to denote the component to which the group in question applies. Thus, the record layout supports structured objects.</p> <p>This secures independence from the sequence of components, because they can be referred to via name.</p> <p>Tags:xml.sequenceOffset=90</p>
swRecord LayoutGroup Axis	AxisIndexType	0..1	attr	<p>This attribute specifies the iteration axis number for a Sw RecordLayoutGroup. The current record layout group then refers exactly to the axis with this number. This means that the values are taken by iterating along the thus referenced axis.</p> <p>Tags:xml.sequenceOffset=30</p>
swRecord LayoutGroup ContentType	SwRecordLayoutGroup Content	0..1	aggr	<p>This is the contents of the recordLayout which is produced for every step of iteration.</p> <p>Tags: xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=100 xml.typeElement=false xml.typeWrapperElement=false</p>
swRecord LayoutGroup From	RecordLayoutIterator Point	0..1	attr	<p>This attribute specifies the iterator index for the point in the axis from which a record layout group is commenced.</p> <p>Negative values are also possible, i.e. the value -4 counts from the fourth value from the end. If this property is missing, the iteration starts with '1'.</p> <p>Tags:xml.sequenceOffset=60</p>





Class	SwRecordLayoutGroup			
swRecordLayoutGroupIndex	NameToken	0..1	attr	This attribute attributes a symbolic name to the iterator of the superimposed record layout group. This can be referenced as a loop index in contained SwRecordLayoutV elements. Tags: xml.sequenceOffset=40
swRecordLayoutGroupStep	Integer	0..1	attr	This attribute specifies the step width for the iterator index that is used for the current record layout group. Note that negative values are also possible, in case of the starting point is higher than the endpoint. If the property is missing, the step width is "1". Tags: xml.sequenceOffset=80
swRecordLayoutGroupTo	RecordLayoutIteratorPoint	0..1	attr	This attribute specifies the end point for the iteration. Negative values are also possible, i.e. the value -4 counts up to the fourth value from the end. If this property is not there, the iteration ends at "-1" which is the last element. Note that depending on the arraySizeSemantics of SwTextProps the iteration ends at the value specified in swMaxTextSize. Tags: xml.sequenceOffset=70

Table E.110: SwRecordLayoutGroup

Class	SwSystemconst			
Package	M2::MSR::DataDictionary::SystemConstant			
Note	This element defines a system constant which serves an input to select a particular variation point. In particular a system constant serves as an operand of the binding function (swSyscond) in a Variation point. Note that the binding process can only happen if a value was assigned to to the referenced system constants. Tags: atp.recommendedPackage=SwSystemconst			
Base	ARElement , ARObject , AtpDefinition , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
swDataDefProps	SwDataDefProps	0..1	aggr	This denotes the data definition properties of the system constant. This supports to express the limits and optionally a conversion within the internal to physical values by a compu method. Tags: xml.sequenceOffset=40

Table E.111: SwSystemconst

Class	SwSystemconstValue			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class assigns a particular value to a system constant.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note





Class		SwSystemconstValue		
annotation	Annotation	*	aggr	This provides the ability to add information why the value is set like it is. Tags: xml.sequenceOffset=30
swSystemconst	SwSystemconst	1	ref	This is the system constant to which the value applies. Tags: xml.sequenceOffset=10
value	Numerical	1	attr	This is the particular value of a system constant. It is specified as Numerical. Further restrictions may apply by the definition of the system constant. The value attribute defines the internal value of the Sw Systemconst as it is processed in the Formula Language. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table E.112: SwSystemconstValue

Class		SwSystemconstantValueSet		
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class represents the ability to specify a set of system constant values. Tags: atp.recommendedPackage=SwSystemconstantValueSets			
Base	ARElement , AObject , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
sw Systemconstant Value	SwSystemconstValue	*	aggr	This is one particular value of a system constant.

Table E.113: SwSystemconstantValueSet

Class		SwcImplementation		
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcImplementation			
Note	This meta-class represents a specialization of the general Implementation meta-class with respect to the usage in application software. Tags: atp.recommendedPackage=SwcImplementations			
Base	ARElement , AObject , CollectableElement , Identifiable , Implementation , MultilanguageReferrable , PackageableElement , Referrable			
Attribute	Type	Mult.	Kind	Note
behavior	SwcInternalBehavior	1	ref	The internal behavior implemented by this Implementation.
perInstance MemorySize	PerInstanceMemory Size	*	aggr	Allows a definition of the size of the per-instance memory for this implementation. The aggregation of PerInstance MemorySize is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects, in this case PerInstanceMemory. Stereotypes: atpVariation Tags: vh.latestBindingTime=preCompileTime





Class	SwcImplementation			
required RTEVendor	String	0..1	attr	Identify a specific RTE vendor. This information is potentially important at the time of integrating (in particular: linking) the application code with the RTE. The semantics is that (if the association exists) the corresponding code has been created to fit to the vendor-mode RTE provided by this specific vendor. Attempting to integrate the code with another RTE generated in vendor mode is in general not possible.

Table E.114: SwcImplementation

Class	SwcInternalBehavior			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	The SwcInternalBehavior of an AtomicSwComponentType describes the relevant aspects of the software-component with respect to the RTE, i.e. the RunnableEntities and the RTEEvents they respond to.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, InternalBehavior, Multilanguage Referrable, Referrable			
Attribute	Type	Mult.	Kind	Note
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	<p>Defines an AUTOSAR typed memory-block that needs to be available for each instance of the SW-component.</p> <p>This is typically only useful if supportsMultipleInstantiation is set to "true" or if the component defines NVRAM access via permanent blocks.</p> <p>The aggregation of arTypedPerInstanceMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
event	RTEEvent	*	aggr	<p>This is a RTEEvent specified for the particular Swc InternalBehavior.</p> <p>The aggregation of RTEEvent is subject to variability with the purpose to support the conditional existence of RTE events. Note: the number of RTE events might vary due to the conditional existence of PortPrototypes using Data ReceivedEvents or due to different scheduling needs of algorithms.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
exclusiveArea Policy	SwcExclusiveArea Policy	*	aggr	<p>Options how to generate the ExclusiveArea related APIs. When no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=exclusiveAreaPolicy, variationPoint.short Label vh.latestBindingTime=preCompileTime</p>





Class	SwcInternalBehavior			
explicitInterRunnableVariable	VariableDataPrototype	*	aggr	<p>Implement state message semantics for establishing communication among runnables of the same component. The aggregation of explicitInterRunnableVariable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
handleTerminationAndRestart	HandleTerminationAndRestartEnum	1	attr	<p>This attribute controls the behavior with respect to stopping and restarting. The corresponding AtomicSwComponentType may either not support stop and restart, or support only stop, or support both stop and restart.</p>
implicitInterRunnableVariable	VariableDataPrototype	*	aggr	<p>Implement state message semantics for establishing communication among runnables of the same component. The aggregation of implicitInterRunnableVariable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
includedDataTypeSet	IncludedDataTypeSet	*	aggr	<p>The includedDataTypeSet is used by a software component for its implementation.</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=includedDataTypeSet</p>
includedModeDeclarationGroupSet	IncludedModeDeclarationGroupSet	*	aggr	<p>This aggregation represents the included Mode DeclarationGroups</p> <p>Stereotypes: atpSplitable Tags:atp.Splitkey=includedModeDeclarationGroupSet</p>
instantiationDataDefProps	InstantiationDataDefProps	*	aggr	<p>The purpose of this is that within the context of a given SwComponentType some data def properties of individual instantiations can be modified. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of Port Prototypes and component local memories like "per InstanceParameter" or "arTypedPerInstanceMemory".</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=instantiationDataDefProps, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
perInstanceMemory	PerInstanceMemory	*	aggr	<p>Defines a per-instance memory object needed by this software component. The aggregation of PerInstanceMemory is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p>





Class	SwcInternalBehavior			
				<p style="text-align: center;">△</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
perInstanceParameter	ParameterData Prototype	*	aggr	<p>Defines parameter(s) or characteristic value(s) that needs to be available for each instance of the software-component. This is typically only useful if supportsMultipleInstantiation is set to "true". The aggregation of perInstanceParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
portAPIOption	PortAPIOption	*	aggr	<p>Options for generating the signature of port-related calls from a runnable to the RTE and vice versa. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=portAPIOption, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
runnable	RunnableEntity	*	aggr	<p>This is a RunnableEntity specified for the particular Swc InternalBehavior.</p> <p>The aggregation of RunnableEntity is subject to variability with the purpose to support the conditional existence of RunnableEntities. Note: the number of RunnableEntities might vary due to the conditional existence of Port Prototypes using DataReceivedEvents or due to different scheduling needs of algorithms.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
serviceDependency	SwcServiceDependency	*	aggr	<p>Defines the requirements on AUTOSAR Services for a particular item.</p> <p>The aggregation of SwcServiceDependency is subject to variability with the purpose to support the conditional existence of ports as well as the conditional existence of ServiceNeeds.</p> <p>The SwcServiceDependency owned by an SwcInternalBehavior can be located in a different physical file in order to support that SwcServiceDependency might be provided in later development steps or even by different expert domain (e.g OBD expert for Obd related Service Needs) tools. Therefore the aggregation is <<atp Splitable>>.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>





Class	SwcInternalBehavior			
shared Parameter	ParameterData Prototype	*	aggr	Defines parameter(s) or characteristic value(s) shared between SwComponentPrototypes of the same Sw ComponentType The aggregation of sharedParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
supports Multiple Instantiation	Boolean	1	attr	Indicate whether the corresponding software-component can be multiply instantiated on one ECU. In this case the attribute will result in an appropriate component API on programming language level (with or without instance handle).
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation. Stereotypes: atpSplitable Tags: atp.Splitkey=shortName

Table E.115: SwcInternalBehavior

Class	SymbolProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This meta-class represents the ability to attach with the symbol attribute a symbolic name that is conform to C language requirements to another meta-class, e.g. AtomicSwComponentType, that is a potential subject to a name clash on the level of RTE source code.			
Base	ARObject , ImplementationProps , Referrable			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

Table E.116: SymbolProps

Enumeration	TcpRoleEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	This enumeration defines whether a TCP node has the tcp server role or the client role.
Literal	Description
connect	Connects the client to a remote TCP host. Tags: atp.EnumerationLiteralIndex=0
listen	Socket is put into the server mode (listen for connections). Tags: atp.EnumerationLiteralIndex=1

Table E.117: TcpRoleEnum

Class	TextValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	The purpose of TextValueSpecification is to define the labels that correspond to enumeration values.			
Base	<i>ARObject</i> , <i>ValueSpecification</i>			
Attribute	Type	Mult.	Kind	Note
value	VerbatimString	1	attr	This is the value itself. Note that vt uses the operator to separate the values for the different bitfield masks in case that the semantics of the related DataPrototype is described by means of a BITFIELD_TEXTTABLE in the associated CompuMethod.

Table E.118: TextValueSpecification

Class	TransformationComSpecProps (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	TransformationComSpecProps holds all the attributes for transformers that are port specific.			
Base	<i>ARObject</i> , <i>Describable</i>			
Subclasses	<i>EndToEndTransformationComSpecProps</i> , <i>UserDefinedTransformationComSpecProps</i>			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table E.119: TransformationComSpecProps

Class	Trigger			
Package	M2::AUTOSARTemplates::CommonStructure::TriggerDeclaration			
Note	A trigger which is provided (i.e. released) or required (i.e. used to activate something) in the given context.			
Base	<i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
swImplPolicy	<i>SwImplPolicyEnum</i>	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers.
triggerPeriod	MultidimensionalTime	0..1	aggr	Optional definition of a period in case of a periodically (time or angle) driven external trigger.

Table E.120: Trigger

Class	TriggerInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A trigger interface declares a number of triggers that can be sent by an trigger source. Tags: atp.recommendedPackage=PortInterfaces			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>PortInterface</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
trigger	<i>Trigger</i>	1..*	aggr	The Trigger of this trigger interface.

Table E.121: TriggerInterface

Class	ValueSpecification (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	<i>ARObject</i>			
Subclasses	<i>AbstractRuleBasedValueSpecification</i> , <i>ApplicationValueSpecification</i> , <i>CompositeValueSpecification</i> , <i>ConstantReference</i> , <i>NotAvailableValueSpecification</i> , NumericalValueSpecification , <i>ReferenceValueSpecification</i> , TextValueSpecification			
Attribute	Type	Mult.	Kind	Note
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table E.122: ValueSpecification

Class	VariableAccess			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	The presence of a VariableAccess implies that a RunnableEntity needs access to a VariableData Prototype. The kind of access is specified by the role in which the class is used.			
Base	<i>ARObject</i> , <i>AbstractAccessPoint</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
accessed Variable	AutosarVariableRef	1	aggr	This denotes the accessed variable.
scope	VariableAccessScope Enum	0..1	attr	This attribute allows for constraining the scope of the corresponding communication. For example, it possible to express whether the communication is intended to cross the boundary of an ECU or whether it is intended not to cross the boundary of a single partition.

Table E.123: VariableAccess

Class	VariableDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A VariableDataPrototype is used to contain values in an ECU application. This means that most likely a VariableDataPrototype allocates "static" memory on the ECU. In some cases optimization strategies might lead to a situation where the memory allocation can be avoided. In particular, the value of a VariableDataPrototype is likely to change as the ECU on which it is used executes.			
Base	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , AutosarDataPrototype , DataPrototype , Identifiable , MultilanguageReferrable , Referrable			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

Table E.124: VariableDataPrototype

Class	VariationPoint			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class represents the ability to express a "structural variation point". The container of the variation point is part of the selected variant if swSyscond evaluates to true and each postBuildVariant Criterion is fulfilled.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
blueprintCondition	DocumentationBlock	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint. Note that variationPoints are not allowed within a blueprintCondition. Tags: xml.sequenceOffset=28
desc	MultiLanguageOverviewParagraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. Tags: xml.sequenceOffset=20
formalBlueprintGenerator	BlueprintGenerator	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint by using ARML. Note that variationPoints are not allowed within a formalBlueprintGenerator. Tags: atp.Status=draft xml.sequenceOffset=30
postBuildVariantCondition	PostBuildVariantCondition	*	aggr	This is the set of post build variant conditions which all shall be fulfilled in order to (postbuild) bind the variation point. Tags: xml.sequenceOffset=40
sdg	Sdg	0..1	aggr	An optional special data group is attached to every variation point. These data can be used by external software systems to attach application specific data. For example, a variant management system might add an identifier, an URL or a specific classifier. Tags: xml.sequenceOffset=50
shortLabel	Identifier	0..1	attr	This provides a name to the particular variation point to support the RTE generator. It is necessary for supporting splittable aggregations and if binding time is later than codeGenerationTime, as well as some RTE conditions. It needs to be unique with in the enclosing Identifiables with the same ShortName. Tags: xml.sequenceOffset=10
swSyscond	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. Note that the multiplicity is 0..1 in order to support pure postBuild variants. Tags: xml.sequenceOffset=30

Table E.125: VariationPoint

Class	ViewMap			
Package	M2::AUTOSARTemplates::GenericStructure::ViewMapSet			
Note	<p>The ViewMap allows to relate any number of elements on the "first" side to any number of elements on the "second" side. Since the ViewMap does not address a specific mapping use-case the roles "first" and "second" shall imply this generality.</p> <p>This mapping allows to trace transformations of artifacts within the AUTOSAR environment. The references to the mapped elements can be plain references and/or InstanceRefs.</p>			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
firstElement	Referrable	*	ref	Reference to identifiable elements on the first "side". Tags: xml.sequenceOffset=20
firstElement Instance	AtpFeature	*	iref	InstanceRefs to elements on the first "side". Tags: xml.sequenceOffset=50
role	Identifier	0..1	attr	This attribute is used to describe specific mapping scenarios, e.g. the mappings: <ul style="list-style-type: none"> AR_AbstractSystemDescription_System Description AR_SystemDescription_SystemExtract Tags: xml.sequenceOffset=10
secondElement	Referrable	*	ref	Reference to identifiable elements on the second "side". Tags: xml.sequenceOffset=30
secondElement Instance	AtpFeature	*	iref	InstanceRefs to elements on the second "side". Tags: xml.sequenceOffset=60

Table E.126: ViewMap

Class	ViewMapSet			
Package	M2::AUTOSARTemplates::GenericStructure::ViewMapSet			
Note	Collection of ViewMaps that are used to establish relationships between different AUTOSAR artifacts. Tags: atp.recommendedPackage=ViewMapSets			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Attribute	Type	Mult.	Kind	Note
viewMap	ViewMap	*	aggr	ViewMaps that are collected by the ViewMapSet.

Table E.127: ViewMapSet

F Splitable Elements in the Scope of this Document

This chapter contains a table of all model elements stereotyped `<<atpSplitable>>` in the scope of this document.

Each entry in Table F.1 consists of the identification of the specific model element itself and the applicable value of the tagged value `atp.Splitkey`.

For more information about the concept of splitable model elements and how these shall be treated please refer to [2].

Name of splitable element	Splitkey
<code>AliasNameSet.aliasName</code>	<code>shortLabel</code>
<code>ClientIdDefinitionSet.clientIdDefinition</code>	<code>shortName, variationPoint.shortLabel</code>
<code>CommunicationCluster.physicalChannel</code>	<code>shortName, variationPoint.shortLabel</code>
<code>CouplingElement.couplingPort</code>	<code>shortName, variationPoint.shortLabel</code>
<code>DataTransformationSet.dataTransformation</code>	<code>shortName, variationPoint.shortLabel</code>
<code>DataTransformationSet.transformationTechnology</code>	<code>shortName, variationPoint.shortLabel</code>
<code>EthernetCluster.couplingPortConnection</code>	<code>couplingPortConnection, variationPoint.shortLabel</code>
<code>EthernetPhysicalChannel.networkEndpoint</code>	<code>shortName</code>
<code>FlatInstanceDescriptor.rtePluginProps</code>	<code>rtePluginProps</code>
<code>FlatMap.instance</code>	<code>shortName, variationPoint.shortLabel</code>
<code>ISignal.dataTransformation</code>	<code>dataTransformation, variationPoint.shortLabel</code>
<code>ISignal.iSignalProps</code>	<code>iSignalProps</code>
<code>ISignalGroup.comBasedSignalGroupTransformation</code>	<code>comBasedSignalGroupTransformation, variationPoint.shortLabel</code>
<code>NmConfig.nmCluster</code>	<code>shortName</code>
<code>NmConfig.nmClusterCoupling</code>	<code>NmClusterCoupling, VariationPoint.shortLabel</code>
<code>NmConfig.nmIfEcu</code>	<code>shortName</code>
<code>PhysicalChannel.frameTriggering</code>	<code>shortName, variationPoint.shortLabel</code>
<code>PhysicalChannel.iSignalTriggering</code>	<code>shortName, variationPoint.shortLabel</code>
<code>PhysicalChannel.pduTriggering</code>	<code>shortName, variationPoint.shortLabel</code>
<code>RootSwCompositionPrototype.calibrationParameter-ValueSet</code>	<code>calibrationParameterValueSet</code>
<code>RootSwCompositionPrototype.flatMap</code>	<code>flatMap</code>
<code>System.j1939SharedAddressCluster</code>	<code>shortName, variationPoint.shortLabel</code>
<code>System.mapping</code>	<code>shortName, variationPoint.shortLabel</code>
<code>System.rootSoftwareComposition</code>	<code>shortName, variationPoint.shortLabel</code>
<code>System.systemDocumentation</code>	<code>shortName, variationPoint.shortLabel</code>
<code>SystemMapping.applicationPartitionToEcuPartition-Mapping</code>	<code>shortName, variationPoint.shortLabel</code>
<code>SystemMapping.cryptoServiceMapping</code>	<code>shortName, variationPoint.shortLabel</code>
<code>SystemMapping.swcToApplicationPartitionMapping</code>	<code>shortName, variationPoint.shortLabel</code>

Table F.1: Usage of splitable elements

G Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped `<<atpVariation>>` in the scope of this document.

Each entry in Table G.1 consists of the identification of the model element itself and the applicable value of the tagged value `vh.latestBindingTime`.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to [2].

Variation Point	Latest Binding Time
<code>AbstractCanCluster</code>	<code>postBuild</code>
<code>AbstractCanCommunicationController</code>	<code>postBuild</code>
<code>AbstractServiceInstance.capabilityRecord</code>	<code>postBuild</code>
<code>AbstractServiceInstance.methodActivationRoutingGroup</code>	<code>postBuild</code>
<code>AliasNameSet.aliasName</code>	<code>preCompileTime</code>
<code>BusMirrorChannel.channel</code>	<code>systemDesignTime</code>
<code>BusMirrorChannelMapping.targetPduTriggering</code>	<code>postBuild</code>
<code>CanCluster</code>	<code>postBuild</code>
<code>CanCommunicationController</code>	<code>postBuild</code>
<code>CanTpConfig.tpAddress</code>	<code>postBuild</code>
<code>CanTpConfig.tpChannel</code>	<code>postBuild</code>
<code>CanTpConfig.tpConnection</code>	<code>postBuild</code>
<code>CanTpConfig.tpEcu</code>	<code>postBuild</code>
<code>CanTpConfig.tpNode</code>	<code>postBuild</code>
<code>ClientIdDefinitionSet.clientIdDefinition</code>	<code>postBuild</code>
<code>ClientIdRange.lowerLimit</code>	<code>postBuild</code>
<code>ClientIdRange.upperLimit</code>	<code>postBuild</code>
<code>CommunicationCluster</code>	<code>postBuild</code>
<code>CommunicationCluster.physicalChannel</code>	<code>systemDesignTime</code>
<code>CommunicationConnector.ecuCommPortInstance</code>	<code>postBuild</code>
<code>CommunicationController</code>	<code>postBuild</code>
<code>ConsumedEventGroup.eventMulticastAddress</code>	<code>postBuild</code>
<code>ConsumedEventGroup.sdClientTimerConfig</code>	<code>postBuild</code>
<code>ConsumedProvidedServiceInstanceGroup.consumedServiceInstance</code>	<code>postBuild</code>
<code>ConsumedProvidedServiceInstanceGroup.providedServiceInstance</code>	<code>postBuild</code>
<code>ConsumedServiceInstance.consumedEventGroup</code>	<code>postBuild</code>
<code>ConsumedServiceInstance.localUnicastAddress</code>	<code>postBuild</code>
<code>ConsumedServiceInstance.remoteUnicastAddress</code>	<code>postBuild</code>
<code>ConsumedServiceInstance.sdClientTimerConfig</code>	<code>postBuild</code>
<code>CouplingElement.couplingPort</code>	<code>postBuild</code>
<code>DataTransformationSet.dataTransformation</code>	<code>codeGenerationTime</code>
<code>DataTransformationSet.transformationTechnology</code>	<code>codeGenerationTime</code>
<code>EcuInstance.associatedConsumedProvidedServiceInstanceGroup</code>	<code>postBuild</code>
<code>EcuInstance.commController</code>	<code>postBuild</code>
<code>EcuInstance.connector</code>	<code>postBuild</code>
<code>EcuInstance.ecuInstanceProps</code>	<code>postBuild</code>
<code>EndToEndTransformationISignalProps</code>	<code>postBuild</code>
<code>EthernetCluster</code>	<code>postBuild</code>
<code>EthernetCluster.couplingPortConnection</code>	<code>postBuild</code>
<code>EthernetCommunicationController</code>	<code>postBuild</code>
<code>EventHandler.eventMulticastAddress</code>	<code>postBuild</code>

EventHandler.sdServerEgTimingConfig	postBuild
FlatMap.instance	postBuild
FlexrayArTpConfig.tpAddress	postBuild
FlexrayArTpConfig.tpChannel	postBuild
FlexrayArTpConfig.tpNode	postBuild
FlexrayCluster	postBuild
FlexrayCommunicationController	postBuild
FlexrayTpConfig.pduPool	postBuild
FlexrayTpConfig.tpAddress	postBuild
FlexrayTpConfig.tpConnection	postBuild
FlexrayTpConfig.tpConnectionControl	postBuild
FlexrayTpConfig.tpEcu	postBuild
FlexrayTpConfig.tpNode	postBuild
Frame.pduToFrameMapping	postBuild
FrameTriggering.pduTriggering	postBuild
Gateway.frameMapping	postBuild
Gateway.iPduMapping	postBuild
Gateway.signalMapping	postBuild
GlobalTimeDomain.gateway	postBuild
GlobalTimeDomain.globalTimeDomainProperty	postBuild
GlobalTimeDomain.globalTimeMaster	postBuild
GlobalTimeDomain.globalTimeSubDomain	postBuild
GlobalTimeDomain.pduTriggering	postBuild
GlobalTimeDomain.slave	postBuild
ISignal.dataTransformation	codeGenerationTime
ISignalGroup.comBasedSignalGroupTransformation	codeGenerationTime
ISignalIPdu.iPduTimingSpecification	postBuild
ISignalIPdu.iSignalToPduMapping	postBuild
ISignalIPdu.pduCounter	preCompileTime
ISignalIPdu.pduReplication	preCompileTime
ISignalIPduGroup.iSignalIPdu	postBuild
ISignalIPduGroup.nmPdu	postBuild
J1939Cluster	postBuild
J1939TpConfig.tpAddress	postBuild
J1939TpConfig.tpConnection	postBuild
J1939TpConfig.tpNode	postBuild
LinCluster	postBuild
LinCommunicationController	postBuild
LinMaster	postBuild
LinPhysicalChannel.scheduleTable	postBuild
LinSlave	postBuild
LinTpConfig.tpAddress	postBuild
LinTpConfig.tpConnection	postBuild
LinTpConfig.tpNode	postBuild
MultiplexedIPdu.dynamicPart	postBuild
MultiplexedIPdu.staticPart	postBuild
NmCluster.nmNode	postBuild
NmConfig.nmCluster	postBuild
NmConfig.nmClusterCoupling	postBuild
NmConfig.nmIfEcu	preCompileTime
PdurIPduGroup.iPdu	postBuild
PduTriggering.iSignalTriggering	postBuild
PhysicalChannel.commConnector	postBuild

PhysicalChannel.frameTriggering	postBuild
PhysicalChannel.iSignalTriggering	postBuild
PhysicalChannel.pduTriggering	postBuild
PncMapping.pncConsumedProvidedServiceInstanceGroup	postBuild
ProvidedServiceInstance.eventHandler	postBuild
ProvidedServiceInstance.localUnicastAddress	postBuild
ProvidedServiceInstance.remoteUnicastAddress	postBuild
ProvidedServiceInstance.sdServerTimerConfig	postBuild
ServiceInstanceCollectionSet.serviceInstance	postBuild
SoAdConfig.connection	postBuild
SoAdConfig.connectionBundle	postBuild
SoAdConfig.socketAddress	postBuild
SocketAddress.staticSocketConnection	postBuild
SOMEIPTransformationISignalProps	postBuild
StaticSocketConnection.ipDuidentifier	postBuild
StaticSocketConnection.remoteAddress	postBuild
System.fibexElement	postBuild
System.j1939SharedAddressCluster	postBuild
System.mapping	postBuild
System.rootSoftwareComposition	systemDesignTime
System.systemDocumentation	systemDesignTime
SystemMapping.applicationPartitionToEcuPartitionMapping	postBuild
SystemMapping.comManagementMapping	systemDesignTime
SystemMapping.cryptoServiceMapping	postBuild
SystemMapping.dataMapping	postBuild
SystemMapping.ecuResourceMapping	systemDesignTime
SystemMapping.mappingConstraint	systemDesignTime
SystemMapping.pncMapping	systemDesignTime
SystemMapping.resourceEstimation	systemDesignTime
SystemMapping.signalPathConstraint	systemDesignTime
SystemMapping.swcToApplicationPartitionMapping	postBuild
SystemMapping.swImplMapping	preCompileTime
SystemMapping.swMapping	preCompileTime
TransformationISignalProps	postBuild
TransformationTechnology.transformationDescription	postBuild
TtcanCluster	postBuild
TtcanCommunicationController	postBuild
UserDefinedCluster	postBuild
UserDefinedCommunicationController	postBuild
UserDefinedTransformationISignalProps	postBuild

Table G.1: Usage of variation points