Document Title	Collection of constraints on AUTOSAR M1 models
Document Category	Technical report (TR)
Document UID	635
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2025-11-27	R25-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2024-11-27	R24-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2023-11-23	R23-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2022-11-24	R22-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2021-11-25	R21-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2020-11-30	R20-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents • Removed all SWS constraints • Split document into 3 documents, reflecting the standards CP, AP, FO
2019-11-28	R19-11	AUTOSAR Release Management	• Updated constraints according to changes in SWS and TPS documents • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Completion of constraint context by adding tables and classtables referenced by model constraints to this document

△

2017-12-08	4.3.1	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2015-07-31	4.2.2	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Updated constraints according to changes in SWS and TPS documents
2013-03-15	4.1.1	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 Document Information and Content 6

2 Autosar Model Constraints 7
2.1 CP_TPS_BSWModuleDescriptionTemplate . . . 7
2.2 CP_TPS_DiagnosticExtractTemplate . . . 40
2.3 CP_TPS_ECUConfiguration . . . 78
2.4 CP_TPS_ECUResourceTemplate . . . 89
2.5 CP_TPS_SoftwareComponentTemplate . . . 90
2.6 CP_TPS_SystemTemplate . . . 244
2.7 CP_TPS_TimingExtensions . . . 416
2.8 FO_TPS_GenericStructureTemplate . . . 437
2.9 FO_TPS_StandardizationTemplate . . . 447

A Mentioned Class Tables 451

B Change history of AUTOSAR traceable items 981
B.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 981
B.1.1 Added Constraints in R26-11 . . . 981
B.1.2 Changed Constraints in R26-11 . . . 984
B.1.3 Deleted Constraints in R26-11 . . . 990
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 993
B.2.1 Added Constraints in R25-11 . . . 993
B.2.2 Changed Constraints in R25-11 . . . 997
B.2.3 Deleted Constraints in R25-11 . . . 999
B.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 1002
B.3.1 Added Constraints in R24-11 . . . 1002
B.3.2 Changed Constraints in R24-11 . . . 1005
B.3.3 Deleted Constraints in R24-11 . . . 1013

References

[1] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[2] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[3] ISO 14229-1 – Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
https://www.iso.org

[4] ISO 26021-2 – Road vehicles – End-of-life activation of on-board pyrotechnic devices – Part 2: Communication requirements
https://www.iso.org

[5] ISO 17356-4: Road vehicles – Open interface for embedded automotive applications – Part 4: OSEK/VDX Communication (COM)

[6] Collection of blueprints for AUTOSAR M1 models
AUTOSAR_FO_MOD_GeneralBlueprints

[7] ISO 10646:2012 – Information technology – Universal Coded Character Set (UCS)
https://www.iso.org

[8] Specification of COM Based Transformer
AUTOSAR_CP_SWS_COMBasedTransformer

[9] SAE J1939-21 Data Link Layer

[10] Transport Layer Security (TLS) Parameters
https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

[11] Standardized M1 Models used for the Definition of AUTOSAR
AUTOSAR_FO_MOD_GeneralDefinitions

[12] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate

[13] XML Path language (XPath)
http://www.w3.org/TR/xpath/

[14] ASAM MCD-2 MC (ASAP2 / A2L)
http://www.asam.net
ASAM_AE_MCD-2_MC_BS_V1-7-1.pdf

1 Document Information and Content

This auxiliary document provides a collection of constraints for AUTOSAR models. All constraints are copied from template specification from the AUTOSAR Classic Platform, so this document does not introduce any new constraints.

A list of the documents that the constraints originate from can be found in the table of contents. Chapter 2 contains the collected constraints, grouped by source documents. All constraints from the same source document are contained within a single section.

2 Autosar Model Constraints

2.1 CP_TPS_BSWModuleDescriptionTemplate

[constr_1275] Applicability of reference `startsOnEvent` for `BswScheduleEvent`

Imposition time: IT_BswMD

⌈The reference BswScheduleEvent.startsOnEvent shall only refer to a BswSchedulableEntity.⌋

[constr_1276] Applicability of reference `startsOnEvent` for `BswOperationInvokedEvent`

Imposition time: IT_BswMD

⌈The reference BswOperationInvokedEvent.startsOnEvent shall only refer to a BswCalledEntity.⌋

[constr_4013] BSW service identifier

Imposition time: IT_BswMD

⌈For Standardized Interfaces, this identifier is defined in the AUTOSAR Software Specification (SWS) of the module. In case the C-function prototype represented by the entry is not standardized, it still can be used optionally, but its value shall differ from the standardized ones.⌋

[constr_4014] Call type and execution context

Imposition time: IT_BswMD

⌈Within a given BswModuleEntry, the following constraint holds for its attributes:

- if attribute callType is set to value interrupt, it is not allowed that attribute executionContext is set to either of the values task or hook
- if attribute callType is set to value scheduled, it is not allowed that attribute executionContext is set to either of the values interruptCat1 or interruptCat2

⌋

[constr_4015] `calledEntry` constraints for direct calls

Imposition time: IT_BswMD

⌈The following holds if callPoint is aggregated as an instance of BswDirectCallPoint:

- BswModuleEntity.callPoint.calledEntry.executionContext shall be identical to BswModuleEntity.implementedEntry.executionContext.
- BswModuleEntity.callPoint.calledEntry.callType shall have the value regular or callback.

⌋

[constr_4016] `BswCalledEntity` constraints

Imposition time: `IT_BswMD`

⌈

- `BswCalledEntity.implementedEntry.callType` shall be `'regular'` or `'callback'`
- `BswCalledEntity.implementedEntry.executionContext` is in general not restricted, but see [constr_4076] for constraints on the server side of a Client-Server communication.

⌋

[constr_4017] `BswSchedulableEntity` constraints

Imposition time: `IT_BswMD`

⌈

- `BswModuleEntity.implementedEntry.callType` shall be `'scheduled'`
- `BswModuleEntity.implementedEntry.executionContext` shall be `'task'`

⌋

[constr_4018] `BswInterruptEntity` constraints

Imposition time: `IT_BswMD`

⌈

- `BswInterruptEntity.implementedEntry.callType` shall be `'interrupt'`
- `BswInterruptEntity.implementedEntry.executionContext` shall be `'interruptCat1'` if and only if `BswInterruptEntity.interruptCategory` is `'Cat1'`
- `BswInterruptEntity.implementedEntry.executionContext` shall be `'interruptCat2'` if and only if `BswInterruptEntity.interruptCategory` is `'Cat2'`
- A `BswInterruptEvent` shall only trigger a `BswInterruptEntity` where attribute `interruptCategory` is set to `BswInterruptCategory.cat2`.

⌋

[constr_4019] BSW module identifier

Imposition time: IT_BswMD

⌈BswModuleDescription.moduleId shall refer to the identifier of the standardized AUTOSAR modules according to [1], if applicable[1]. Otherwise (e.g. for ICC2 clusters) the identifier shall either be empty or chosen differently from the ones given in [1].⌋

[constr_4020] Allowed categories of BswModuleDescription

Imposition time: IT_BswMD

⌈

category	*Explanation*
BSW_MODULE	Specifies a single BSW module (ICC3 granularity).
BSW_CLUSTER	Specifies a BSW module cluster (ICC2 granularity).
LIBRARY	Specifies a Library (not restricted to be used within the BSW).

⌋

[constr_4021] Implementation policy of function pointer target

Imposition time: IT_BswMD

⌈
A BswModuleEntry can only be used as target of a function pointer (SwPointerTargetProps.functionPointerSignature), if its swServiceImplPolicy is 'standard'.⌋

[constr_4022] BswModuleEntity only uses the module's interface

Imposition time: IT_BswMD

⌈

- BswModuleEntity.implementedEntry shall refer to an element declared as implementedEntry of the enclosing BswModuleDescription
- BswModuleEntity.callPoint.calledEntry - where callPoint is instantiated from BswDirectCallPoint - shall refer to an element declared as expectedEntry or implementedEntry of the enclosing BswModuleDescription.
- BswModuleEntity.callPoint.calledEntry - where callPoint is instantiated from BswSynchronousServerCallPoint or BswAsynchronousServerCallPoint - shall refer to an element declared as requiredClientServerEntry of the enclosing BswModuleDescription.
- BswModuleEntity.callPoint - where callPoint is instantiated from BswAsynchronousServerCallResultPoint - shall refer to an Bsw-

[1]Note that there may be more than one module in an ECU software with the same identifier, e.g. according to the standard Complex Drivers all have the same identifier.

AsynchronousServerCallPoint declared in turn as callPoint of the same BswModuleEntity.

- BswModuleEntity.issuedTrigger shall refer to an element declared as releasedTrigger of the enclosing BswModuleDescription
- BswModuleEntity.managedModeGroup shall refer to an element declared as providedModeGroup of the enclosing BswModuleDescription
- BswModuleEntity.accessedModeGroup shall refer to an element declared as requiredModeGroup of the enclosing BswModuleDescription
- BswModuleEntity.dataSendPoint.accessedVariable shall refer to an element declared as providedData of the enclosing BswModuleDescription
- BswModuleEntity.dataReceivePoint.accessedVariable shall refer to an element declared as requiredData of the enclosing BswModuleDescription
- an accessedModeGroup should be allowed to refer to an element declared as providedModeGroup

⌋

[constr_4023] External trigger shall belong to the interface

Imposition time: IT_BswMD

⌈A BswExternalTriggerOccurredEvent shall refer to a Trigger that is declared via BswModuleDescription.requiredTrigger for the same module.⌋

[constr_4024] Semantics of BSW mode switch event

Imposition time: IT_BswMD

⌈If BswModeSwitchEvent.activation has the value onTransition BswModeSwitchEvent shall refer to two different modes belonging to the same instance of ModeDeclarationGroup, their order defining the direction of the transition. In all other cases, BswModeSwitchEvent shall refer to exactly one mode.⌋

[constr_4025] Modes used by BSW mode switch event

Imposition time: IT_BswMD

⌈The ModeDeclaration used by BswModeSwitchEvent shall belong to the ModeDeclarationGroupPrototype referred as BswInternalBehavior.entity.accessedModeGroup of the enclosing BswInternalBehavior.⌋

[constr_4026] Mode group used by BSW mode switch acknowledge event

Imposition time: IT_BswMD

⌈The ModeDeclarationGroupPrototype used by BswModeSwitchedAckEvent shall be referred as BswModuleDescription.providedModeGroup by the same module.⌋

[constr_4028] Semantics of memory section type

Imposition time: IT_BswMD

⌈sectionType shall be semantically compatible to the usage of the enclosing SwAddrMethod, this means especially that if SwAddrMethod is associated by ExecutableEntity-s, the sectionType shall be usable as code section, if it is associated by SwDataDefProps, sectionType shall be usable as data section.⌋

[constr_4029] Measured stack usage

Imposition time: IT_BswMD

⌈The attribute values of MeasuredStackUsage shall fulfill:
minimumMemoryConsumption <= averageMemoryConsumption <= maximum-MemoryConsumption⌋

[constr_4030] Measured heap usage

Imposition time: IT_BswMD

⌈The attribute values of MeasuredHeapUsage shall fulfill:
minimumMemoryConsumption <= averageMemoryConsumption <= maximum-MemoryConsumption⌋

[constr_4031] Analyzed execution time

Imposition time: IT_BswMD

⌈The attribute values of AnalyzedExecutionTime shall fulfill:
bestCaseExecutionTime <= bestCaseExecutionTime⌋

[constr_4032] Measured execution time

Imposition time: IT_BswMD

⌈The attribute values of MeasuredExecutionTime shall fulfill:
minimumExecutionTime <= nominalExecutionTime <= maximumExecution-Time⌋

[constr_4033] Simulated execution time

Imposition time: IT_BswMD

⌈The attribute values of SimulatedExecutionTime shall fulfill:
minimumExecutionTime <= nominalExecutionTime <= maximumExecution-Time⌋

[constr_4034] Target and context of MC emulation reference

Imposition time: IT_BswMD

⌈Within one ImplementationElementInParameterInstanceRef, the target shall refer to a sub-element of the ParameterDataPrototype which is referred as context.⌋

[constr_4038] `bswModuleDependency` shall refer to a different module

Imposition time: IT_BswMD

⌈

- BswModuleDescription.bswModuleDependency.targetModuleId (if given) shall differ from BswModuleDescription.moduleId. This does not hold if the value is 254 (used for IO Hardware Abstraction modules) or 255 (used for Complex Driver modules).
- BswModuleDependency.targetModuleRef (if given) shall differ from the package location of the BswModuleDescription that owns the BswModuleDependency.

⌋

[constr_4039] Semantics of `SwcBswMapping`

Imposition time: IT_BswMD

⌈An SwcBswMapping is only valid, if the referred SwcInternalBehavior is aggregated by a ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.⌋

[constr_4040] Synchronized mode groups shall have same type

Imposition time: IT_BswMD

⌈SwcBswSynchronizedModeGroupPrototype can only refer to equally typed ModeDeclarationGroupPrototypes, i.e. which have identical ModeDeclarationGroups.⌋

[constr_4041] Synchronized mode groups shall have same context

Imposition time: IT_BswMD

⌈The mapping defined by SwcBswSynchronizedModeGroupPrototype implies that the component providing the one mode group prototype is also mapped to the module which provides the other mode group prototype by means of synchronizing their respective behaviors in SwcBswMapping.⌋

[constr_4042] Synchronized triggers shall have same context

Imposition time: IT_BswMD

⌈The mapping defined by SwcBswSynchronizedTrigger implies that the component providing the one trigger is also mapped to the module which provides the other trigger by means of synchronizing their respective behaviors in SwcBswMapping.⌋

[constr_4043] Period of `BswTimingEvent`

Imposition time: IT_BswMD

⌈BswTimingEvent.period shall be greater than 0.⌋

[constr_4044] Content of McSwEmulationMethodSupport

Imposition time: IT_BswMD

⌈The following constraints hold for the attributes of McSwEmulationMethodSupport:

- If category is DOUBLE_POINTERED, a baseReference shall exist.
- If category is SINGLE_POINTERED, a referenceTable shall exist.
- If category is INITIALIZED_RAM, one or more elementGroups shall exist.

⌋

[constr_4045] implementationConfigVariant of preconfigured configuration

Imposition time: IT_BswMD

⌈An EcucModuleConfigurationValues element with the implementationConfigVariant set to the value PreconfiguredConfiguration shall only be referenced in the role preconfiguredConfiguration and no other value for implementationConfigVariant is allowed in this role.⌋

[constr_4046] implementationConfigVariant of recommended configuration

Imposition time: IT_BswMD

⌈An EcucModuleConfigurationValues element with the implementationConfigVariant set to the value RecommendedConfiguration shall only be referenced in the role recommendedConfiguration and no other value for implementationConfigVariant is allowed in this role.⌋

[constr_4047] Multiplicity of vendor specific configuration parameters

Imposition time: IT_BswMD

⌈The association BswImplementation.vendorSpecificModuleDef shall be implemented as reference to one or more instances of EcucModuleDef if the underlying BswModuleDescription has the category BSW_CLUSTER. In all other cases, it shall refer to exactly one instance of EcucModuleDef (the one belonging to this module).⌋

[constr_4048] Multiplicity of preconfigured values

Imposition time: IT_BswMD

⌈The association BswImplementation.preconfiguredConfiguration shall be implemented as reference to zero or more different instances of EcucModuleConfigurationValues if the underlying BswModuleDescription has the category BSW_CLUSTER. In all other cases, it shall refer to at most one instance of EcucModuleConfigurationValues (the one belonging to this module).⌋

[constr_4051] RoleBasedDataAssignment in BSW

Imposition time: IT_BswMD

⌈When used in the context of BswServiceDependency, the following restriction hold for date references described by RoleBasedDataAssignment:

- Within RoleBasedDataAssignment.usedDataElement, only the reference AutosarVariableRef.localVariable is applicable.
- Within RoleBasedDataAssignment.usedParameterElement, only the reference AutosarParameterRef.localParameter is applicable.
- The reference RoleBasedDataAssignment.usedPim shall not be set.

⌋

[constr_4052] BswModuleEntry returnType direction

Imposition time: IT_BswMD

⌈
BswModuleEntry.returnType.direction shall not have the value **in** or **inout**.⌋

[constr_4053] BswModuleEntry argument direction

Imposition time: IT_BswMD

⌈
If BswModuleEntry.argument.direction has the value **out** or **inout**, the corresponding BswModuleEntry.argument.swDataDefProps plus eventually referred ImplementationDataType shall be such that they result in a pointer declaration.⌋

[constr_4054] Unambiguous links to addressing method

Imposition time: IT_BswMD

⌈MemorySection.executableEntity shall not be defined, if MemorySection.swAddrMethod represents a data section. MemorySection.executableEntity shall not refer to an ExecutableEntity which is linked to a different SwAddrMethod than MemorySection.swAddrMethod.⌋

[constr_4056] BswModuleEntry with no returnType

Imposition time: IT_BswMD

⌈
In case of an empty return type (“void” in C) the reference BswModuleEntry.returnType shall not be set.⌋

[constr_4057] BswModuleEntry with no argument

Imposition time: IT_BswMD

⌈
In case of an empty argument list (“void” in C) no reference BswModuleEntry.argument shall be set.⌋

[constr_4058] Different mode groups in mapped BSWM and SWC shall have different names

Imposition time: IT_BswMD

⌈If an SwcInternalBehavior is mapped to a BswInternalBehavior the corresponding SWC and BSW module descriptions may not refer to different ModeDeclarationGroups having the same shortName but different elements. This holds especially if these mode groups are not synchronized but used independently.⌋

[constr_4059] Different mode groups referred by a BSWM shall have different names

Imposition time: IT_BswMD

⌈A BswModuleDescription may not refer to different ModeDeclarationGroups (via requiredModeGroup and/or providedModeGroup) having the same shortName but different elements.⌋

[constr_4060] Allowed values of Trigger.swImplPolicy for BSW

Imposition time: IT_BswMD

⌈The **only** allowed values for the attribute Trigger.swImplPolicy are either STANDARD (in which case the Trigger processing does not use a queue) or QUEUED (in which case the processing of Triggers positively uses a queue).⌋

[constr_4061] Completeness of MC emulation reference

Imposition time: IT_BswMD

⌈If an McDataInstance in the role of a subElement of another McDataInstance specifies an instanceInMemory, then the containing McDataInstance shall also specify an instanceInMemory.

The target of the latter (i.e. upper level) instanceInMemory shall be identical (including array index, if defined) to the context of the first (i.e. lower level) instanceInMemory.⌋

[constr_4062] Mandatory symbol for McDataInstance root

Imposition time: IT_BswMD

⌈McDataInstances directly aggregated in McSupportData shall have a valid McDataInstance.symbol.⌋

[constr_4063] Restrictions of ModeRequestTypeMap in BSW

Imposition time: IT_BswMD

⌈For every ModeDeclarationGroup referenced by a ModeDeclarationGroupPrototype used in a BswModuleDescription a ModeRequestTypeMap shall exist that points to the ModeDeclarationGroup and also to an eligible ImplementationDataType.

The ModeRequestTypeMap shall be aggregated by a DataTypeMappingSet which is referenced from the BswInternalBehavior that is aggregated by the BswModuleDescription.⌋

[constr_4064] Synchronized triggers shall implement same policy

Imposition time: IT_BswMD

⌈The mapping defined by SwcBswSynchronizedTrigger is only valid if the attribute SwcBswSynchronizedTrigger.swcTrigger.swImplPolicy has the same value as the attribute SwcBswSynchronizedTrigger.bswTrigger.swImplPolicy.⌋

[constr_4065] Allowed values of BswInternalTriggeringPoint.swImplPolicy

Imposition time: IT_BswMD

⌈The **only** allowed values for the attribute BswInternalTriggeringPoint.swImplPolicy are either STANDARD (in which case the internal trigger processing does not use a queue) or QUEUED (in which case the internal trigger processing uses a queue).⌋

[constr_4066] BswModeSwitchEvent and the definition of ModeTransition

Imposition time: IT_BswMD

⌈For each pair of ModeDeclarations referenced by a BswModeSwitchEvent with attribute activation set to onTransition a ModeTransition shall be defined in the corresponding direction (i.e. from exitedMode to enteredMode). This constraint shall only apply if the respective ModeDeclarationGroup defines at least one modeTransition.⌋

[constr_4068] McFunctionDataRefSet.flatMapEntry's semantic

Imposition time: IT_BswMD

⌈

- An McFunctionDataRefSet aggregated in the role of McFunction.defCalprmSet or McFunction.refCalprmSet shall only refer to FlatInstanceDescriptors that
  - either can be traced down to a ParameterDataPrototype
  - or can be traced down to a VariableDataPrototype of category COM_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, CUBE_5 or VAL_BLK

  and which are declared for calibration access i.e. have an associated SwDataDefProps.swCalibrationAccess set to readWrite or readOnly.
- An McFunctionDataRefSet aggregated in the role of McFunction.inMeasurementSet, McFunction.outMeasurementSet or McFunction.locMeasurementSet shall only refer to FlatInstanceDescriptors that can be traced down to either a VariableDataPrototype, an Argument-

DataPrototype or a ModeDeclarationGroupPrototype and are declared as measurable i.e. have an associated SwDataDefProps.swCalibrationAccess set to readOnly.

⌋

[constr_4069] McFunctionDataRefSet.mcDataInstance's semantic

Imposition time: IT_BswMD

⌈

- An McFunctionDataRefSet aggregated in the role of McFunction.defCalprmSet or McFunction.refCalprmSet shall only refer to McDataInstances that are declared for calibration access i.e. are aggregated in the role McSupportData.mcParameterInstance or McSupportData.mcVariableInstance of category COM_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, CUBE_5 or VAL_BLK.
- An McFunctionDataRefSet aggregated in the role of McFunction.inMeasurementSet, McFunction.outMeasurementSet or McFunction.locMeasurementSet shall only refer to McDataInstances that are declared as measurable i.e. are aggregated in the role McSupportData.mcVariableInstance.

⌋

[constr_4070] Applicability of BswModuleEntity.activationReason

Imposition time: IT_BswMD

⌈An activationReason shall not be set

- for instances of BswInterruptEntity
- for instances of BswCalledEntity

⌋

[constr_4071] Synchronized runnables and schedulable entities shall be consistent

Imposition time: IT_BswMD

⌈A SwcBswRunnableMapping that maps a RunnableEntity to a BswCalledEntity or BswSchedulableEntity is only valid if several attributes of the mapped RunnableEntity and BswSchedulableEntity are consistent, especially all of the following constraints apply to the attributes of the given instance of SwcBswRunnableMapping:

- swcRunnable.symbol shall be identical to the symbol of bswEntity as defined in [TPS_BSWMDT_04138].

- swcRunnable.minimumStartInterval shall be identical to bswEntity.minimumStartInterval.
- swcRunnable.canBeInvokedConcurrently shall be identical to bswEntity.implementedEntry.isReentrant.
- swcRunnable.swAddrMethod shall either be empty or shall have identical attributes as the SwAddrMethod defined via bswEntity.swAddrMethod. This is required to ensure a unique configuration for the memory segment of the underlying code entity.
- swcRunnable.activationReason and bswEntity.activationReason shall have identical shortName if they define the same bitPosition and shall have identical bitPosition if they define the same shortName

⌋

[constr_4072] Constraints of SectionNamePrefix.implementedIn

Imposition time: IT_BswMD

⌈

- The SectionNamePrefix and the DependencyOnArtifact connected via this link shall belong to the same BswImplementation.
- The DependencyOnArtifact referred by this link shall be aggregated by BswImplementation in the role requiredArtifact.
- The DependencyOnArtifact referred by this link shall have the category value set to MEMMAP.

⌋

[constr_4073] McDataAccessDetails shall refer to one ECU Extract

Imposition time: IT_BswMD

⌈Within one given McDataAccessDetails, all instances of System referenced as the base of any McDataAccessDetails.variableAccess or as the base of any McDataAccessDetails.rteEvent shall be identical and of category ECU_EXTRACT.⌋

[constr_4074] Compatibility of BswModuleClientServerEntry-s

Imposition time: IT_BswMD

⌈Two BswModuleClientServerEntry-s are compatible if and only if all of the following conditions hold:

- Their synchronicity values are identical. These values are taken from the attribute isSynchronous or, if this is undefined, from encapsulatedEntry.isSynchronous.

- The two BswModuleEntry-s referred as encapsulatedEntry have SwServiceArg, returnType, serviceId and swServiceImplPolicy identical.

⌋

[constr_4075] Constraints for providedData and requiredData

Imposition time: IT_BswMD

⌈Sender-Receiver communication in BSW is restricted to the pattern of so-called *explicit communication* (in the same way as described for software components in [2]) with queued behavior. This leads to some constraints for the VariableDataPrototype referred in the role BswModuleDescription.providedData or BswModuleDescription.requiredData:

- It shall not have an initValue.
- Its swDataDefProps.swImplPolicy shall be set to queued.
- Its swDataDefProps.swCalibrationAccess shall be set to notAccessible.

There are no further formal constraints on the attributes of the VariableDataPrototype to be used in these roles or on the underlying AutosarDataPrototype.⌋

[constr_4076] Constraints on BswModuleEntry used for Client-Server

Imposition time: IT_BswMD

⌈A BswModuleEntry used in the role BswModuleClientServerEntry.encapsulatedEntry shall have attribute values as follows:

- callType shall be regular or callback.
- executionContext shall be task.

⌋

[constr_4077] Constraints for BswModuleEntity.reentrancyLevel

Imposition time: IT_BswMD

⌈

- If the attribute isReentrant of a BswModuleEntry referred by an BswModuleEntity in the role implementedEntry has the value true, then the attribute reentrancyLevel of the same BswModuleEntity (if it exists) can only have the values singleCoreReentrant or multicoreReentrant.
- If the attribute isReentrant of a BswModuleEntry referred by an BswModuleEntity in the role implementedEntry has the values false, then there are no restrictions for the values of the attribute reentrancyLevel of the same BswModuleEntity (if it exists).

⌋

[constr_4078] Consistent usage of `BswOperationInvokedEvent`

Imposition time: IT_BswMD

⌈The BswCalledEntity referred by the attribute BswOperationInvokedEvent.startsOnEvent shall refer to the same BswModuleEntry (via its attribute implementedEntry) as the BswOperationInvokedEvent (via its attribute entry.encapsulatedEntry).⌋

[constr_4079] `calledEntry` constraints for client-server calls

Imposition time: IT_BswMD

⌈

- The BswModuleClientServerEntry aggregated as calledEntry in a BswSynchronousServerCallPoint shall have the attribute isSynchronous = true.
- The BswModuleClientServerEntry aggregated as calledEntry in a BswAsynchronousServerCallPoint shall have the attribute isSynchronous = false.

⌋

[constr_4080] Existence of reception policy

Imposition time: IT_BswMD

⌈If a VariableDataPrototype is referred from a dataReceivePoint of any BswModuleEntity in a given BswInternalBehavior, then exactly one corresponding BswDataReceptionPolicy shall by aggregated by this BswInternalBehavior.⌋

[constr_4081] Mode group used by BSW mode manager error event

Imposition time: IT_BswMD

⌈The ModeDeclarationGroupPrototype used by BswModeManagerErrorEvent shall be referred as BswModuleDescription.providedModeGroup by the same module.⌋

[constr_4083] `BswDistinguishedPartition` shall be used only in the context of a particular `BswInternalBehavior`

Imposition time: IT_BswMD

⌈All instances of BswEvent, BswModuleCallPoint and BswVariableAccess which refer to a BswDistinguishedPartition shall belong to the same BswInternalBehavior that also aggregates the referred BswDistinguishedPartition.⌋

[constr_4084] Consistency of references of `InternalBehavior`

Imposition time: IT_BswMD

⌈The SwcInternalBehavior referenced by SwcBswMapping.swcBehavior in the SwcBswMapping determined by SwcImplementation.swcBswMapping shall be identical to the SwcInternalBehavior referenced by SwcImplementation.behavior.⌋

[constr_4085] Consistency of references of `InternalBehavior`

Imposition time: IT_BswMD

⌈The BswInternalBehavior referenced by SwcBswMapping.bswBehavior in the SwcBswMapping determined by BswImplementation.swcBswMapping shall be identical to the BswInternalBehavior referenced by BswImplementation.behavior.⌋

[constr_4087] Usage of category "MACRO"

Imposition time: IT_BswMD

⌈
It is only allowed to use the category "MACRO" for SwServiceArg if the owning BswModuleEntry has its swServiceImplPolicy attribute set to macro.⌋

[constr_4088] Existence of `RoleBasedDataTypeAssignment.role` vs. `RoleBasedDataAssignment.role`

Imposition time: IT_BswMD

⌈The usage of a RoleBasedDataTypeAssignment with attribute role set to the value temporaryRamBlock is only allowed if **no** RoleBasedDataAssignment defined with attribute role set to value defaultValue exists in the owning BswServiceDependency.⌋

[constr_4089] Association `callbackHeader` is only applicable for BSW modules

Imposition time: IT_BswMD

⌈The reference Code.callbackHeader is only allowed to be used if the Code is aggregated by a BswImplementation in the role codeDescriptor.⌋

[constr_4090] The `callbackHeader` reference has to be consistent with behavior reference

Imposition time: IT_BswMD

⌈The reference Code.callbackHeader is only allowed to reference ServiceNeeds in the context of the BswServiceDependency that in turn is aggregated by a BswImplementation via BswInternalBehavior that is owning the Code in the role codeDescriptor.⌋

[constr_4091] `AccessCount.value` needs to be unambiguous

Imposition time: IT_BswMD

⌈AUTOSAR model shall define at most one `AccessCount.value` per `countProfile` for a specific `AbstractAccessPoint`.⌋

[constr_4092] Number of `ErrorTracerNeeds` in `BswInternalBehavior`

Imposition time: IT_BswMD

⌈A BswInternalBehavior shall provide at most one `ErrorTracerNeeds` element.⌋

[constr_4093] Entries linked to `BswModuleEntry`s shall have compatible signature

Imposition time: IT_BswMD

⌈Matching `BswModuleEntry`s according to [TPS_BSWMDT_04130] are compatible if the following conditions are fulfilled:

- both or neither of them define a `returnType`
- when the `returnType`s are defined, the `SwServiceArg`s in the role `returnType` shall be compatible
- both define the same number of compatible arguments in same order

⌋

[constr_4094] compatibility of `SwServiceArg` in role `returnType`

Imposition time: IT_BswMD

⌈`SwServiceArg` in role `returnType` are compatible if they are identically typed⌋

[constr_4095] Compatibility of `SwServiceArg` in role `argument`

Imposition time: IT_BswMD

⌈`SwServiceArg` in role `returnType` are compatible if:

- they are identically typed

and

- if both do have the same `shortName`

⌋

[constr_4096] Matching `BswModuleEntry`s should have compatible attributes

Imposition time: IT_BswMD

⌈Matching `BswModuleEntry`s according to [TPS_BSWMDT_04130] should be defined with identical values of the attributes

- `callType`
- `executionContext`

- isReentrant
- isSynchronous
- serviceId
- swServiceImplPolicy
- bswEntryKind

⌋

[constr_4097] Limitation on the number of BswExclusiveAreaPolicys

Imposition time: IT_BswMD

⌈An ExclusiveArea can only be referenced by at most one BswExclusiveAreaPolicy.⌋

[constr_4098] No mode disabling for BswOperationInvokedEvent

Imposition time: IT_BswMD

⌈A BswOperationInvokedEvent shall not have a reference to a ModeDeclaration in the role disabledInMode.⌋

[constr_4099] Support of multiple instantiation

Imposition time: IT_BswMD

⌈If a BSW Module supports multiple instantiation the attribute vendorApiInfix is mandatory.⌋

[constr_4100] Uniqueness of module implementation prefixes

Imposition time: IT_BswMD

⌈Inside one ECU the Module implementation prefixes (Mip) of BSW Modules shall be unique.⌋

[constr_4101] Semantics of McGroupDataRefSet.flatMapEntry

Imposition time: IT_BswMD

⌈

- An McGroupDataRefSet aggregated in the role of McGroup.refCalprmSet or McGroup.refCalprmSet shall only refer to FlatInstanceDescriptors that can either be traced down to a ParameterDataPrototype or can be traced down to a VariableDataPrototype of category COM_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, CUBE_5 or VAL_BLK and which are declared for calibration access i.e. have an associated SwDataDefProps.swCalibrationAccess set to readWrite or readOnly.
- An McGroupDataRefSet aggregated in the role of McGroup.refMeasurementSet shall only refer to FlatInstanceDescriptors that can be traced down to either a VariableDataPrototype, an Argument-

DataPrototype or a ModeDeclarationGroupPrototype and are declared as measurable i.e. have an associated SwDataDefProps.swCalibrationAccess set to readOnly.

⌋

[constr_4102] Semantics of McGroupDataRefSet.mcDataInstance

Imposition time: IT_BswMD

⌈

- An McGroupDataRefSet aggregated in the role of McGroup.refCalprmSet shall only refer to McDataInstances that are declared for calibration access i.e. are aggregated in the role McSupportData.mcParameterInstance or McSupportData.mcParameterInstance of category VALUE, COM_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, CUBE_5 or VAL_BLK.
- An McGroupDataRefSet aggregated in the role of McGroup.refMeasurementSet shall only refer to McDataInstances that are declared as measurable i.e. are aggregated in the role McSupportData.mcVariableInstance.

⌋

[constr_4103] Name convention for SectionNamePrefix.symbol

Imposition time: IT_BswMD

⌈In case a BSW module is split into allocatable memory parts the existing (according to [SWS_MemMap_00041]) SectionNamePrefix.symbol shall be set in the <MIP>_<FEATURE> form, where:

- <MIP>: is the capitalized module implementation prefix
- <FEATURE>: is the name of the sub-feature in the BSW module denoting the allocatable memory part

⌋

[constr_4104] Referencing of MemorySections to SectionNamePrefix

Imposition time: IT_BswMD

⌈In case a BSW module or Software Component is split into allocatable memory parts all MemorySections belonging to the same allocatable memory part shall reference the identical SectionNamePrefix representing the allocatable memory part.⌋

[constr_4105] Use of attribute task or cat2Isr

Imposition time: IT_BswMD

⌈Only one of the attributes is allowed to exist. Either task or cat2Isr should be configured.⌋

[constr_4106] Restriction for the value of SwServiceArg.swImplPolicy

Imposition time: IT_BswMD

⌈The attribute SwServiceArg.swImplPolicy shall only have one of the following values:

- SwImplPolicyEnum.const
- SwImplPolicyEnum.standard

⌋

[constr_4107] swImplPolicy for SwServiceArg

Imposition time: IT_BswMD

⌈The overriding value of attribute swImplPolicy of a SwServiceArg shall be standard or const.⌋

[constr_4108] Restriction regarding the value of SwServiceArg.category

Imposition time: IT_BswMD

⌈The attribute SwServiceArg.category shall only have the following values:

- VALUE[2]
- DATA_REFERENCE
- FUNCTION_REFERENCE
- TYPE_REFERENCE
- MACRO

⌋

[constr_9316] Multi instantiated BSW Modules not mappable

Imposition time: IT_BswMD

⌈In case a BSW Module is multi instantiated in an ECU its BswImplementations shall not reference a SwcBswMapping in the role swcBswMapping.⌋

[2]This option has **very few** valid use cases, e.g. for defining a function pointer in native C notation, for example: `int (*SwCluC_BManif_VoidFncPtrType)(void);`

[constr_9357] Existence of attributes of McDataInstance depending on the category

Imposition time: IT_BswMD

⌈

Attributes of McDataInstance	Attribute Existence per McDataInstance.category													
	VALUE	VAL_BLK	STRUCTURE	UNION	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
arraySize					1									
displayIdentifier	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
role	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
symbol	1	1	1	1	1	1	1	1	1	1	1	1	1	1
subElement			1..*	1..*	1..*									
flatMapEntry	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
instanceInMemory	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
mcDataAccessDetails	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
mcDataAssignment	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*
resultingRptSwPrototyping-Access	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
rptImplPolicy	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.compu-Method	0..1	0..1					0..1			0..1	0..1	0..1	0..1	0..1
resultingProperties.unit	0..1	0..1								0..1	0..1	0..1	0..1	0..1
resultingProperties.dataConstr	0..1	0..1			0..1		0..1			0..1	0..1	0..1	0..1	0..1
resultingProperties.swCalprm-AxisSet								0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.swData-Dependency	0..1						0..1			0..1	0..1	0..1	0..1	0..1
resultingProperties.swHost-Variable														
resultingProperties.swBit-Representation														
resultingProperties.baseType	0..1													
resultingProperties.swRecord-Layout	0..1	0..1				0..1		1	1	1	1	1	1	1
resultingProperties.additional-NativeTypeQualifier	0..1		0..1	0..1	0..1	0..1		0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.stepSize	0..1	0..1			0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.display-Presentation	0..1	0..1			0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.swAlign-ment														
resultingProperties.swImpl-Policy														
resultingProperties.swAddr-Method														
resultingProperties.swIntended-Resolution														

▽

△

Attributes of McDataInstance	Attribute Existence per McDataInstance.category													
	VALUE	VAL_BLK	STRUCTURE	UNION	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
resultingProperties.swPointer-TargetProps														
resultingProperties.swInterpolation-Method								0..1	0..1	0..1	0..1	0..1	0..1	0..1
resultingProperties.swIs-Virtual														
resultingProperties.swValue-BlockSize		1												
resultingProperties.swValue-BlockSizeMult		1												
resultingProperties.annotation	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*	0..*
resultingProperties.display-Format	0..1	0..1	0..1	0..1	0..1	0..1	0..1			0..1	0..1	0..1	0..1	0..1
resultingProperties.implementation-DataType														
resultingProperties.invalid-Value	0..1		0..1		0..1	0..1								
resultingProperties.swText-Props						1								

⌋

[constr_10257] Existence of attribute BswServiceDependency.serviceNeeds

Imposition time: IT_BswMD

⌈For each BswServiceDependency, the attribute serviceNeeds shall exist.⌋

[constr_10258] Existence of the reference in the role RoleBasedBswModule-EntryAssignment.assignedEntry

Imposition time: IT_BswMD

⌈For each RoleBasedBswModuleEntryAssignment, the reference in the role assignedEntry shall exist.⌋

[constr_10259] Existence of attribute RoleBasedBswModuleEntry-Assignment.role

Imposition time: IT_BswMD

⌈For each RoleBasedBswModuleEntryAssignment, the attribute role shall exist.⌋

[constr_10260] Existence of attribute BswModuleEntry.callType

Imposition time: IT_BswMD

⌈For each BswModuleEntry, the attribute callType shall exist.⌋

[constr_10261] Existence of attribute BswModuleEntry.executionContext

Imposition time: IT_BswMD

⌈For each BswModuleEntry, the attribute executionContext shall exist.⌋

[constr_10262] Existence of attribute BswModuleEntry.isReentrant

Imposition time: IT_BswMD

⌈For each BswModuleEntry, the attribute isReentrant shall exist.⌋

[constr_10263] Existence of attribute BswModuleEntry.isSynchronous

Imposition time: IT_BswMD

⌈For each BswModuleEntry, the attribute isSynchronous shall exist.⌋

[constr_10264] Existence of attribute BswModuleEntry.swServiceImplPolicy

Imposition time: IT_BswMD

⌈For each BswModuleEntry, the attribute swServiceImplPolicy shall exist.⌋

[constr_10265] Existence of attribute BswEntryRelationshipSet.bswEntryRelationship

Imposition time: IT_BswMD

⌈For each BswEntryRelationshipSet, the attribute bswEntryRelationship shall exist at least once.⌋

[constr_10266] Existence of attribute BswEntryRelationship.bswEntryRelationshipType

Imposition time: IT_BswMD

⌈For each BswEntryRelationship, the attribute bswEntryRelationshipType shall exist.⌋

[constr_10267] Existence of reference in the role BswEntryRelationship.from

Imposition time: IT_BswMD

⌈For each BswEntryRelationship, the reference in the role from shall exist.⌋

[constr_10268] Existence of reference in the role BswEntryRelationship.to

Imposition time: IT_BswMD

⌈For each BswEntryRelationship, the reference in the role to shall exist.⌋

[constr_10269] Existence of the reference in the role BswModuleClientServerEntry.encapsulatedEntry

Imposition time: IT_BswMD

⌈For each BswModuleClientServerEntry, the the reference in the role encapsulatedEntry shall exist.⌋

[constr_10270] Existence of attribute AccessCountSet.countProfile

Imposition time: IT_BswMD

⌈For each AccessCountSet, the attribute countProfile shall exist.⌋

[constr_10271] Existence of attribute AccessCount.value

Imposition time: IT_BswMD

⌈For each AccessCount, the attribute value shall exist.⌋

[constr_10272] Existence of the reference in the role BswModuleEntity.implementedEntry

Imposition time: IT_BswMD

⌈For each BswModuleEntity, the reference in the role implementedEntry shall exist.⌋

[constr_10273] Existence of attribute BswInterruptEntity.interruptCategory

Imposition time: IT_BswMD

⌈For each BswInterruptEntity, the attribute interruptCategory shall exist.⌋

[constr_10274] Existence of attribute BswInterruptEntity.interruptSource

Imposition time: IT_BswMD

⌈For each BswInterruptEntity, the attribute interruptSource shall exist.⌋

[constr_10275] Existence of the reference in the role BswDirectCallPoint.calledEntry

Imposition time: IT_BswMD

⌈For each BswDirectCallPoint, the reference in the role calledEntry shall exist.⌋

[constr_10276] Existence of the reference in the role BswSynchronousServerCallPoint.calledEntry

Imposition time: IT_BswMD

⌈For each BswSynchronousServerCallPoint, the reference in the role calledEntry shall exist.⌋

[constr_10277] Existence of the reference in the role BswAsynchronousServerCallPoint.calledEntry

Imposition time: IT_BswMD

⌈For each BswAsynchronousServerCallPoint, the reference in the role calledEntry shall exist.⌋

[constr_10278] Existence of the reference in the role BswAsynchronousServerCallResultPoint.asynchronousServerCallPoint

Imposition time: IT_BswMD

⌈For each BswAsynchronousServerCallResultPoint, the reference in the role asynchronousServerCallPoint shall exist.⌋

[constr_10279] Existence of the reference in the role BswVariableAccess.accessedVariable

Imposition time: IT_BswMD

⌈For each BswVariableAccess, the reference in the role accessedVariable shall exist.⌋

[constr_10280] Existence of the reference in the role BswExclusiveAreaPolicy.exclusiveArea

Imposition time: IT_BswMD

⌈For each BswExclusiveAreaPolicy, the reference in the role exclusiveArea shall exist.⌋

[constr_10281] Existence of attribute BswTimingEvent.period

Imposition time: IT_BswMD

⌈For each BswTimingEvent, the attribute period shall exist.⌋

[constr_10282] Existence of the reference in the role BswInternalTriggerOccurredEvent.eventSource

Imposition time: IT_BswMD

⌈For each BswInternalTriggerOccurredEvent, the reference in the role eventSource shall exist.⌋

[constr_10283] Existence of the reference in the role BswExternalTriggerOccurredEvent.trigger

Imposition time: IT_BswMD

⌈For each BswExternalTriggerOccurredEvent, the reference in the role trigger shall exist.⌋

[constr_10284] Existence of attribute BswModeSwitchEvent.activation

Imposition time: IT_BswMD

⌈For each BswModeSwitchEvent, the attribute activation shall exist.⌋

[constr_10285] Existence of the reference in the role BswModeSwitchedAckEvent.modeGroup

Imposition time: IT_BswMD

⌈For each BswModeSwitchedAckEvent, the reference in the role modeGroup shall exist.⌋

[constr_10286] Existence of the reference in the role BswModeManagerErrorEvent.modeGroup

Imposition time: IT_BswMD

⌈For each BswModeManagerErrorEvent, the reference in the role modeGroup shall exist.⌋

[constr_10287] Existence of the reference in the role BswOperationInvokedEvent.entry

Imposition time: IT_BswMD

⌈For each BswOperationInvokedEvent, the reference in the role entry shall exist.⌋

[constr_10288] Existence of the reference in the role BswAsynchronousServerCallReturnsEvent.eventSource

Imposition time: IT_BswMD

⌈For each BswAsynchronousServerCallReturnsEvent, the reference in the role eventSource shall exist.⌋

[constr_10289] Existence of the reference in the role BswDataReceivedEvent.data

Imposition time: IT_BswMD

⌈For each BswDataReceivedEvent, the reference in the role data shall exist.⌋

[constr_10290] Existence of the reference in the role BswTriggerDirectImplementation.masteredTrigger

Imposition time: IT_BswMD

⌈For each BswTriggerDirectImplementation, the reference in the role masteredTrigger shall exist.⌋

[constr_10291] Existence of the reference in the role BswModeSenderPolicy.providedModeGroup

Imposition time: IT_BswMD

⌈For each BswModeSenderPolicy, the reference in the role providedModeGroup shall exist.⌋

[constr_10292] Existence of attribute BswModeSenderPolicy.queueLength

Imposition time: IT_BswMD

⌈For each BswModeSenderPolicy, the attribute queueLength shall exist.⌋

[constr_10293] Existence of attribute BswModeSwitchAckRequest.timeout

Imposition time: IT_BswMD

⌈For each BswModeSwitchAckRequest, the attribute timeout shall exist.⌋

[constr_10294] Existence of the reference in the role BswModeReceiverPolicy.requiredModeGroup

Imposition time: IT_BswMD

⌈For each BswModeReceiverPolicy, the reference in the role requiredModeGroup shall exist.⌋

[constr_10295] Existence of attribute BswModeReceiverPolicy.supportsAsynchronousModeSwitch

Imposition time: IT_BswMD

⌈For each BswModeReceiverPolicy, the attribute supportsAsynchronousModeSwitch shall exist.⌋

[constr_10296] Existence of reference in the role BswDataReceptionPolicy.receivedData

Imposition time: IT_BswMD

⌈For each BswDataReceptionPolicy, the reference in the role receivedData shall exist.⌋

[constr_10297] Existence of attribute BswQueuedDataReceptionPolicy.queueLength

Imposition time: IT_BswMD

⌈For each BswQueuedDataReceptionPolicy, the attribute queueLength shall exist.⌋

[constr_10298] Existence of the reference in the role SwcBswRunnableMapping.bswEntity

Imposition time: IT_BswMD

⌈For each SwcBswRunnableMapping, the reference in the role bswEntity shall exist.⌋

[constr_10299] Existence of the reference in the role SwcBswRunnableMapping.swcRunnable

Imposition time: IT_BswMD

⌈For each SwcBswRunnableMapping, the reference in the role swcRunnable shall exist.⌋

[constr_10300] Existence of the reference in the role SwcBswSynchronizedTrigger.bswTrigger

Imposition time: IT_BswMD

⌈For each SwcBswSynchronizedTrigger, the reference in the role bswTrigger shall exist.⌋

[constr_10301] Existence of the instanceRef in the role SwcBswSynchronizedTrigger.swcTrigger

Imposition time: IT_BswMD

⌈For each SwcBswSynchronizedTrigger, the instanceRef in the role swcTrigger shall exist.⌋

[constr_10302] Existence of attribute BswImplementation.arReleaseVersion

Imposition time: IT_BswMD

⌈For each BswImplementation, the attribute arReleaseVersion shall exist.⌋

[constr_10303] Existence of the reference in the role BswImplementation.behavior

Imposition time: IT_BswMD

⌈For each BswImplementation, the reference in the role behavior shall exist.⌋

[constr_10304] Existence of attribute DependencyOnArtifact.usage

Imposition time: IT_BswMD

⌈For each DependencyOnArtifact, the attribute usage shall exist at least once.⌋

[constr_10305] Existence of attribute WorstCaseStackUsage.memoryConsumption

Imposition time: IT_BswMD

⌈For each WorstCaseStackUsage, the attribute memoryConsumption shall exist.⌋

[constr_10306] Existence of attribute MeasuredStackUsage.averageMemoryConsumption

Imposition time: IT_BswMD

⌈For each MeasuredStackUsage, the attribute averageMemoryConsumption shall exist.⌋

[constr_10307] Existence of attribute MeasuredStackUsage.maximumMemoryConsumption

Imposition time: IT_BswMD

⌈For each MeasuredStackUsage, the attribute maximumMemoryConsumption shall exist.⌋

[constr_10308] Existence of attribute RoughEstimateStackUsage.memoryConsumption

Imposition time: IT_BswMD

⌈For each RoughEstimateStackUsage, the attribute memoryConsumption shall exist.⌋

[constr_10309] Existence of attribute WorstCaseHeapUsage.memoryConsumption

Imposition time: IT_BswMD

⌈For each WorstCaseHeapUsage, the attribute memoryConsumption shall exist.⌋

[constr_10310] Existence of attribute MeasuredHeapUsage.averageMemoryConsumption

Imposition time: IT_BswMD

⌈For each MeasuredHeapUsage, the attribute averageMemoryConsumption shall exist.⌋

[constr_10311] Existence of attribute MeasuredHeapUsage.maximumMemoryConsumption

Imposition time: IT_BswMD

⌈For each MeasuredHeapUsage, the attribute maximumMemoryConsumption shall exist.⌋

[constr_10312] Existence of attribute RoughEstimateHeapUsage.memoryConsumption

Imposition time: IT_BswMD

⌈For each RoughEstimateHeapUsage, the attribute memoryConsumption shall exist.⌋

[constr_10313] Existence of attribute ExecutionTime.hardwareConfiguration

Imposition time: IT_BswMD

⌈For each ExecutionTime, the attribute hardwareConfiguration shall exist.⌋

[constr_10314] Existence of attribute ExecutionTime.softwareContext

Imposition time: IT_BswMD

⌈For each ExecutionTime, the attribute softwareContext shall exist.⌋

[constr_10315] Existence of attribute HardwareConfiguration.additionalInformation

Imposition time: IT_BswMD

⌈For each HardwareConfiguration, the attribute additionalInformation shall exist.⌋

[constr_10316] Existence of attribute HardwareConfiguration.processorMode

Imposition time: IT_BswMD

⌈For each HardwareConfiguration, the attribute processorMode shall exist.⌋

[constr_10317] Existence of attribute HardwareConfiguration.processorSpeed

Imposition time: IT_BswMD

⌈For each HardwareConfiguration, the attribute processorSpeed shall exist.⌋

[constr_10318] Existence of reference MemorySectionLocation.providedMemory

Imposition time: IT_BswMD

⌈For each MemorySectionLocation, the reference in the role providedMemory shall exist.⌋

[constr_10319] Existence of reference MemorySectionLocation.softwareMemorySection

Imposition time: IT_BswMD

⌈For each MemorySectionLocation, the reference in the role softwareMemorySection shall exist.⌋

[constr_10320] Existence of attribute SoftwareContext.input

Imposition time: IT_BswMD

⌈For each SoftwareContext, the attribute input shall exist.⌋

[constr_10321] Existence of attribute SoftwareContext.state

Imposition time: IT_BswMD

⌈For each SoftwareContext, the attribute state shall exist.⌋

[constr_10323] Existence of attribute AnalyzedExecutionTime.bestCaseExecutionTime

Imposition time: IT_BswMD

⌈For each AnalyzedExecutionTime, the attribute bestCaseExecutionTime shall exist.⌋

[constr_10324] Existence of attribute AnalyzedExecutionTime.worstCaseExecutionTime

Imposition time: IT_BswMD

⌈For each AnalyzedExecutionTime, the attribute worstCaseExecutionTime shall exist.⌋

[constr_10325] Existence of attribute MeasuredExecutionTime.maximumExecutionTime

Imposition time: IT_BswMD

⌈For each MeasuredExecutionTime, the attribute maximumExecutionTime shall exist.⌋

[constr_10326] Existence of attribute MeasuredExecutionTime.minimumExecutionTime

Imposition time: IT_BswMD

⌈For each MeasuredExecutionTime, the attribute minimumExecutionTime shall exist.⌋

[constr_10327] Existence of attribute MeasuredExecutionTime.nominalExecutionTime

Imposition time: IT_BswMD

⌈For each MeasuredExecutionTime, the attribute nominalExecutionTime shall exist.⌋

[constr_10328] Existence of the reference in the role BswEvent.startsOnEvent

Imposition time: IT_BswMD

⌈For each BswEvent, the reference in the role startsOnEvent shall exist.⌋

[constr_10329] Existence of the instanceRef in the role McDataAccessDetails.variableAccess

Imposition time: IT_BswMD

⌈For each McDataAccessDetails, the instanceRef in the role variableAccess shall exist at least once.⌋

[constr_10330] Existence of attribute RptServicePoint.symbol

Imposition time: IT_BswMD

⌈For each RptServicePoint, the attribute symbol shall exist.⌋

[constr_10331] Existence of attribute SimulatedExecutionTime.maximumExecutionTime

Imposition time: IT_BswMD

⌈For each SimulatedExecutionTime, the attribute maximumExecutionTime shall exist.⌋

[constr_10332] Existence of attribute SimulatedExecutionTime.minimumExecutionTime

Imposition time: IT_BswMD

⌈For each SimulatedExecutionTime, the attribute minimumExecutionTime shall exist.⌋

[constr_10333] Existence of attribute SimulatedExecutionTime.nominalExecutionTime

Imposition time: IT_BswMD

⌈For each SimulatedExecutionTime, the attribute nominalExecutionTime shall exist.⌋

[constr_10334] Existence of attribute RoughEstimateOfExecutionTime.additionalInformation

Imposition time: IT_BswMD

⌈For each RoughEstimateOfExecutionTime, the attribute additionalInformation shall exist.⌋

[constr_10335] Existence of attribute RoughEstimateOfExecutionTime.estimatedExecutionTime

Imposition time: IT_BswMD

⌈For each RoughEstimateOfExecutionTime, the attribute estimatedExecutionTime shall exist.⌋

[constr_10336] Existence of the reference in the role SwcBswSynchronizedModeGroupPrototype.bswModeGroup

Imposition time: IT_BswMD

⌈For each SwcBswSynchronizedModeGroupPrototype, the reference in the role bswModeGroup shall exist.⌋

[constr_10337] Existence of the instanceRef in the role SwcBswSynchronizedModeGroupPrototype.swcModeGroup

Imposition time: IT_BswMD

⌈For each SwcBswSynchronizedModeGroupPrototype, the instanceRef in the role swcModeGroup shall exist.⌋

[constr_10338] Existence of attribute MultidimensionalTime.cseCode

Imposition time: IT_BswMD

⌈For each MultidimensionalTime, the attribute cseCode shall exist.⌋

[constr_10339] Existence of attribute MultidimensionalTime.cseCodeFactor

Imposition time: IT_BswMD

⌈For each MultidimensionalTime, the attribute cseCodeFactor shall exist.⌋

[constr_10340] Existence of attribute McSwEmulationMethodSupport.category

Imposition time: IT_BswMD

⌈For each McSwEmulationMethodSupport, the attribute category shall exist.⌋

[constr_10341] Existence of attribute McSwEmulationMethodSupport.shortLabel

Imposition time: IT_BswMD

⌈For each McSwEmulationMethodSupport, the attribute shortLabel shall exist.⌋

[constr_10342] Existence of the reference in the role McParameterElementGroup.ramLocation

Imposition time: IT_BswMD

⌈For each McParameterElementGroup, the reference in the role ramLocation shall exist.⌋

[constr_10343] Existence of the reference in the role McParameterElementGroup.romLocation

Imposition time: IT_BswMD

⌈For each McParameterElementGroup, the reference in the role romLocation shall exist.⌋

[constr_10344] Existence of attribute McParameterElementGroup.shortLabel

Imposition time: IT_BswMD

⌈For each McParameterElementGroup, the attribute shortLabel shall exist.⌋

[constr_10345] Existence of the reference in the role ImplementationElementInParameterInstanceRef.context

Imposition time: IT_BswMD

⌈For each ImplementationElementInParameterInstanceRef, the reference in the role context shall exist.⌋

[constr_10346] Existence of the reference in the role ImplementationElementInParameterInstanceRef.target

Imposition time: IT_BswMD

⌈For each ImplementationElementInParameterInstanceRef, the reference in the role target shall exist.⌋

[constr_10347] Existence of the instanceRef in the role McDataAccessDetails.rteEvent

Imposition time: IT_BswMD

⌈For each McDataAccessDetails, the instanceRef in the role rteEvent shall exist at least once.⌋

[constr_10349] Existence of attribute RptSupportData.executionContext

Imposition time: IT_BswMD

⌈For each RptSupportData, the attribute executionContext shall exist at least once.⌋

[constr_10350] Existence of attribute RptSupportData.rptComponent

Imposition time: IT_BswMD

⌈For each RptSupportData, the attribute rptComponent shall exist at least once.⌋

[constr_10351] Existence of attribute RptSupportData.rptServicePoint

Imposition time: IT_BswMD

⌈For each RptSupportData, the attribute rptServicePoint shall exist at least once.⌋

[constr_10352] Existence of attribute RptComponent.rptExecutableEntity

Imposition time: IT_BswMD

⌈For each RptComponent, the attribute rptExecutableEntity shall exist at least once.⌋

[constr_10353] Existence of attribute RptExecutableEntity.rptExecutableEntityEvent

Imposition time: IT_BswMD

⌈For each RptExecutableEntity, the attribute rptExecutableEntityEvent shall exist at least once.⌋

[constr_10354] Existence of attribute RptExecutableEntity.symbol

Imposition time: IT_BswMD

⌈For each RptExecutableEntity, the attribute symbol shall exist.⌋

[constr_10355] Existence of the reference in the role RptExecutableEntityEvent.executionContext

Imposition time: IT_BswMD

⌈For each RptExecutableEntityEvent, the reference in the role executionContext shall exist at least once.⌋

[constr_10356] Existence of attribute RptExecutableEntityEvent.rptEventId

Imposition time: IT_BswMD

⌈For each RptExecutableEntityEvent, the attribute rptEventId shall exist.⌋

[constr_10357] Existence of attribute RptExecutableEntityEvent.rptExecutableEntityProperties

Imposition time: IT_BswMD

⌈For each RptExecutableEntityEvent, the attribute rptExecutableEntityProperties shall exist.⌋

[constr_10358] Existence of the reference in the role RptExecutableEntityEvent.rptServicePointPost

Imposition time: IT_BswMD

⌈For each RptExecutableEntityEvent, the reference in the role rptServicePointPost shall exist at least once.⌋

[constr_10359] Existence of the reference in the role RptExecutableEntityEvent.rptServicePointPre

Imposition time: IT_BswMD

⌈For each RptExecutableEntityEvent, the reference in the role rptServicePointPre shall exist at least once.⌋

[constr_10360] Existence of attribute RptServicePoint.serviceId

Imposition time: IT_BswMD

⌈For each RptServicePoint, the attribute serviceId shall exist.⌋

[constr_10362] Existence of attribute AliasNameSet.aliasName

Imposition time: IT_BswMD

⌈For each AliasNameSet, the attribute aliasName shall exist at least once.⌋

[constr_10363] Existence of attribute AliasNameAssignment.shortLabel

Imposition time: IT_BswMD

⌈For each AliasNameAssignment, the attribute shortLabel shall exist.⌋

2.2 CP_TPS_DiagnosticExtractTemplate

[constr_1324] Existence of DiagnosticDataIdentifier.representsVin

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Within the context of a given DiagnosticContributionSet, the attribute DiagnosticDataIdentifier.representsVin shall have the value true for only a single DiagnosticDataIdentifier.⌋

[constr_1325] Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈

Attributes of SwDataDefProps	DiagnosticDataElement.swDataDefProps
additionalNativeTypeQualifier	
annotation	
baseType.baseTypeDefinition.baseTypeEncoding	D
baseType.baseTypeDefinition.baseTypeSize	D
baseType.baseTypeDefinition.byteOrder	D
baseType.baseTypeDefinition.memAlignment	
baseType.baseTypeDefinition.nativeDeclaration	
compuMethod	D
dataConstr	D

▽

△

Attributes of SwDataDefProps	DiagnosticDataElement.swDataDefProps
displayFormat	D
displayPresentation	
implementationDataType	
invalidValue	
swAddrMethod	
swAlignment	
swBitRepresentation	
swCalibrationAccess	
swCalprmAxisSet	
swComparisonVariable	
swDataDependency	
swImplPolicy	
swIntendedResolution	
swInterpolationMethod	
swIsVirtual	
swPointerTargetProps	
swRecordLayout	
swRefreshTiming	
swTextProps	
swValueBlockSize	
unit	D
valueAxisDataType	

⌋

[constr_1326] Existence of a variable-sized array

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of the attribute DiagnosticDataElement.arraySizeSemantics **shall not** be set to ArraySizeSemanticsEnum.variableSize if the respective DiagnosticDataElement is referenced from a DiagnosticServiceDataMapping.⌋

[constr_1327] Multiplicity of DiagnosticEcuInstanceProps.ecuInstance

Imposition time: CP: IT_DiagDes

⌈The multiplicity of DiagnosticEcuInstanceProps.ecuInstance shall be limited to 1 and the enclosing DiagnosticContributionSet shall only refer to at most one DiagnosticEcuInstanceProps **if** the enclosing DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT.⌋

[constr_1328] Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceTable referenced by any given DiagnosticContributionSet in the role serviceTable shall define a reference in the role DiagnosticServiceTable.ecuInstance to an EcuInstance that is also referenced in the role DiagnosticEcuInstanceProps.ecuInstance by

a DiagnosticEcuInstanceProps referenced by the mentioned DiagnosticContributionSet **if** the respective DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT.⌋

[constr_1329] Existence of concrete sub-classes of DiagnosticServiceClass in the context created by a DiagnosticContributionSet

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈One of the following mutually exclusive conditions shall apply for the existence of any concrete sub-class of DiagnosticServiceClass in the context created by a DiagnosticContributionSet:

- The subclass of DiagnosticServiceClass (except DiagnosticCustomServiceClass) shall only appear once in the context created by a DiagnosticContributionSet.
- A DiagnosticCustomServiceClass with a given value of attribute customServiceId shall only appear once in the context created by a DiagnosticContributionSet.
- If the subclass of DiagnosticServiceClass (except DiagnosticCustomServiceClass) appears multiple times in the context created by a DiagnosticContributionSet, then all instances of the sub-class of DiagnosticServiceClass shall have identical values for all of their attributes.

  In the case of aggregations, the number of aggregated elements shall be identical and the values of primitive attributes of aggregated elements shall again be identical.

⌋

[constr_1330] Custom service identifier shall not overlap with standardized service identifiers

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of the attribute DiagnosticCustomServiceClass.customServiceId shall not be set to any of the values reserved for standardized service identifiers as defined by [3, ISO 14229-1].⌋

[constr_1331] Existence of DiagnosticEcuReset.customSubFunctionNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The attribute DiagnosticEcuReset.customSubFunctionNumber shall only exist if the value of DiagnosticEcuReset.category is outside the standardized set of values as defined by [TPS_DEXT_01056].⌋

[constr_1332] Value range for DiagnosticEcuReset.customSubFunction-Number

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The allowed value for DiagnosticEcuReset.customSubFunctionNumber shall always be within the closed interval **0x40** .. **0x7E**.⌋

[constr_1333] Existence of attributes of meta-class DiagnosticMemory-Identifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If a DiagnosticMemoryIdentifier is referenced in the role memoryRange by a DiagnosticRequestDownload or a DiagnosticRequestUpload, then the attributes

- memoryLowAddress
- memoryLowAddressLabel
- memoryHighAddress
- memoryHighAddressLabel
- accessPermission

shall **not** exist for the referenced DiagnosticMemoryIdentifier.⌋

[constr_1334] Existence of DiagnosticComControl.customSubFunction-Number

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The attribute DiagnosticComControl.customSubFunctionNumber shall only exist if the value of DiagnosticComControl.category is outside the standardized set of values as defined by [TPS_DEXT_01057].⌋

[constr_1335] Possible values for DiagnosticComControl.customSub-FunctionNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Given the fulfillment of [constr_1334], the value of a given DiagnosticCom-Control.customSubFunctionNumber shall always be within the closed interval **0x40** .. **0x5F** (for manufacturer-specific sub-functions) or the closed interval **0x60** .. **0x7E** (for supplier-specific sub-functions).⌋

[constr_1336] Applicable value range for DiagnosticComControlSpecific-Channel.subnetNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticComControlSpecificChannel.subnet-Number shall be within the closed interval **1** .. **14**.⌋

[constr_1337] Allowed value range for attribute DiagnosticComControlSubNodeChannel.subNodeNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticComControlSubNodeChannel.subNodeNumber shall not exceed the closed interval **0** .. **65535**.⌋

[constr_1338] Maximum number of aggregated DiagnosticReadDataByPeriodicIDClass.periodicRate

Imposition time: CP: IT_DiagDes

⌈The number of aggregated periodicRate within the context of one DiagnosticReadDataByPeriodicIDClass shall be within the closed interval 1..3.⌋

[constr_1339] Existence of DiagnosticRoutine.start

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈In a complete DiagnosticExtract, the attribute DiagnosticRoutine.start shall always exist for any given DiagnosticRoutine.⌋

[constr_1340] Consistency of DiagnosticServiceSwMapping with respect to synchronously called DiagnosticRoutines

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that only aggregates a DiagnosticStartRoutine in the role start shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.synchronous.⌋

[constr_1341] Consistency of DiagnosticServiceSwMapping with respect to asynchronously called DiagnosticRoutines

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that aggregates a DiagnosticStopRoutine and/or DiagnosticRequestRoutineResults in the role stop or requestResult shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.asynchronous.⌋

[constr_1342] Possible values for DiagnosticSecurityAccess.requestSeedId

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of the attribute DiagnosticSecurityAccess.requestSeedId **shall only be set to an odd number**[3].

[3]The even numbers are reserved for the identification of the corresponding *sendKey* sub-function, as explained by [TPS_DEXT_01036].

The supported value range consists of the following list:

- all odd numbers in the closed interval **0x01** .. **0x41**
- **0x5F** (this corresponds to the case of *end-of-life activation of on-board pyrotechnic devices according to [4, ISO 26021-2]*)
- all odd numbers in the closed interval **0x61** .. **0x7E**

⌋

[constr_1343] Simultaneous existence of the attributes DiagnosticServiceDataMapping.diagnosticDataElement and DiagnosticDataByIdentifier.dataIdentifier

Imposition time: CP: IT_DiagDes

⌈A DiagnosticServiceDataMapping.diagnosticDataElement shall also be aggregated by a DiagnosticDataByIdentifier in the role dataIdentifier.dataElement.dataElement.⌋

[constr_1345] DiagnosticDataElement shall not (finally) be aggregated by a DiagnosticRoutine

Imposition time: CP: IT_DiagDes

⌈A DiagnosticDataElement that is referenced by a DiagnosticServiceDataMapping shall not (finally) be aggregated by a DiagnosticRoutine.⌋

[constr_1346] Allowed values of DiagnosticServiceSwMapping.serviceInstance

Imposition time: CP: IT_DiagDes

⌈The applicability of the DiagnosticServiceSwMapping is limited to a predefined set of diagnostic services.

By regulation of the AUTOSAR standard, DiagnosticServiceSwMapping.serviceInstance shall only point to the following sub-classes of DiagnosticServiceInstance:

- DiagnosticRoutine
- DiagnosticSecurityAccess
- DiagnosticReadDataByIdentifier
- DiagnosticWriteDataByIdentifier
- DiagnosticIOControl

⌋

[constr_1347] Existence of references owned by DiagnosticServiceSwMapping

Imposition time: CP: IT_DiagDes

⌈For any given DiagnosticServiceSwMapping, **one and only one of** the following references shall exist:

- DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency
- DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem
- DiagnosticServiceSwMapping.mappedBswServiceDependency

⌋

[constr_1349] Value of udsDtcValue shall be unique

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of DiagnosticTroubleCodeUds.udsDtcValue shall be unique for all DiagnosticTroubleCodeUds that refer to the same DiagnosticMemoryDestination via the reference DiagnosticTroubleCodeUds.troubleCodeProps.diagnosticMemory.⌋

[constr_1350] Value of DiagnosticTroubleCodeGroup.groupNumber shall be unique

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of DiagnosticTroubleCodeGroup.groupNumber shall be unique to any other DTC and DTC group value.⌋

[constr_1351] Value of DiagnosticTroubleCodeGroup.groupNumber

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈To be compliant to ISO, the value of DiagnosticTroubleCodeGroup.groupNumber shall be set as defined in [3, ISO 14229-1].⌋

[constr_1352] Existence of maxNumberFreezeFrameRecords vs. freezeFrame

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If the attribute DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords exists, then the attribute DiagnosticTroubleCodeProps.freezeFrame shall not exist or vice versa.⌋

[constr_1353] Applicability of [constr_1352]

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈[constr_1352] shall apply in the identical way (either one or the other attribute shall exist) for all DiagnosticTroubleCodeProps within the context of all DiagnosticContributionSets of category DIAGNOSTIC_ECU_EXTRACT that refer to the same EcuInstance.⌋

[constr_1354] Existence of the reference in the role DiagnosticTroubleCodeProps.snapshotRecordContent

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If one of the attributes DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords or DiagnosticTroubleCodeProps.freezeFrame exists then the reference in the role DiagnosticTroubleCodeProps.snapshotRecordContent shall exist.⌋

[constr_1355] Value of extendedDataRecord.recordNumber

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈To be compliant with the ISO standard, the value of extendedDataRecord.recordNumber shall be set in the interval as defined in [3, ISO 14229-1].⌋

[constr_1357] Value of freezeFrame.recordNumber

Imposition time: CP: IT_DiagDes

⌈To be compliant with ISO, the value of freezeFrame.recordNumber shall be set in the interval as defined in [3, ISO 14229-1].⌋

[constr_1359] Condition for the existence of attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Attribute debounceCounterStorage of meta-class DiagnosticDebounceAlgorithmProps shall only exist if the aggregation of attribute debounceAlgorithm at DiagnosticDebounceAlgorithmProps actually aggregates a DiagEventDebounceCounterBased.⌋

[constr_1361] Number of DiagnosticEventToEnableConditionGroupMapping elements per DiagnosticEvent

Imposition time: CP: IT_DiagDes

⌈The mapping element DiagnosticEventToEnableConditionGroupMapping shall be created no more than once per DiagnosticEvent.

If several DiagnosticEventToEnableConditionGroupMapping elements referring to the same DiagnosticEvent are defined, then the Enable Condition Group mapping shall be regarded as defective.⌋

[constr_1362] Number of DiagnosticEventToStorageConditionGroupMapping elements per DiagnosticEvent

Imposition time: CP: IT_DiagDes

⌈The mapping element DiagnosticEventToStorageConditionGroupMapping shall be created no more than once per DiagnosticEvent.

If several DiagnosticEventToStorageConditionGroupMapping elements referring to the same DiagnosticEvent are defined, then the Storage Condition Group mapping shall be regarded as defective.⌋

[constr_1378] Value of DiagnosticMemoryDestinationUserDefined.memoryId

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Within the scope of one DiagnosticContributionSet, no two (or more) DiagnosticMemoryDestinationUserDefineds shall exist that share the same value for attribute DiagnosticMemoryDestinationUserDefined.memoryId⌋

[constr_1379] Existence of DiagnosticMemoryDestinationPrimary

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Within the scope of one DiagnosticContributionSet only one DiagnosticMemoryDestinationPrimary shall exist.⌋

[constr_1394] Value of DiagnosticDataElement.maxNumberOfElements depending on its existence

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If the attribute DiagnosticDataElement.maxNumberOfElements exists, then its value shall be greater than 0.⌋

[constr_1405] Value of DiagnosticProtocol.serviceTable vs. DiagnosticServiceTable.protocolKind

Imposition time: CP: IT_DiagDes

⌈If the reference DiagnosticProtocol.serviceTable exists, then the value of DiagnosticProtocol.protocolKind shall be identical to the value of DiagnosticServiceTable.protocolKind.⌋

[constr_1406] Meta-class DiagnosticServiceTable.diagnosticConnection vs. meta-class DiagnosticProtocol.diagnosticConnection

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticServiceTable exists that fulfills the following conditions:

- reference DiagnosticServiceTable.diagnosticConnection exists
- the DiagnosticServiceTable is referenced by means of DiagnosticProtocol.serviceTable

then all of the DiagnosticConnections referenced by means of DiagnosticServiceTable.diagnosticConnection shall also be referenced in the role diagnosticConnection from a DiagnosticProtocol that in turn references the respective DiagnosticServiceTable in the role DiagnosticProtocol.serviceTable.⌋

[constr_1411] Existence of attribute DiagnosticMemoryIdentifier.memoryHighAddressLabel vs. DiagnosticMemoryIdentifier.memoryHighAddress

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈At most **one** of the attributes in the following list shall exist:

- DiagnosticMemoryIdentifier.memoryHighAddressLabel
- DiagnosticMemoryIdentifier.memoryHighAddress

⌋

[constr_1412] Existence of DiagnosticMemoryIdentifier.memoryLowAddressLabel vs. DiagnosticMemoryIdentifier.memoryLowAddress

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈At most **one** of the attributes in the following list shall exist:

- DiagnosticMemoryIdentifier.memoryLowAddressLabel
- DiagnosticMemoryIdentifier.memoryLowAddress

⌋

[constr_1419] Value of DiagnosticSecurityLevel.accessDataRecordSize

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If the attribute DiagnosticSecurityLevel.accessDataRecordSize exists then its value shall be greater than zero.⌋

[constr_1421] Consistency of attribute DiagnosticDynamicallyDefineDataIdentifierClass.subfunction

Imposition time: CP: IT_DiagDes

⌈The values of DiagnosticDynamicallyDefineDataIdentifierClass.subfunction shall not repeat, i.e. every value of DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum shall at most appear once in the subfunction attribute.⌋

[constr_1435] Debouncing in the presence of a DiagnosticEventPortMapping

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticEventPortMapping exists and the enclosed DiagnosticEventPortMapping.diagnosticEvent is also referenced by a DiagnosticEventToDebounceAlgorithmMapping then the concrete subclass of the respective DiagnosticEventToDebounceAlgorithmMapping.debounceAlgorithm.debounceAlgorithm shall be identical to the DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds.diagEventDebounceAlgorithm.⌋

[constr_1447] Restrictions for the value of DiagnosticParameterIdentifier.id

Imposition time: CP: IT_DiagDes

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticParameterIdentifier.id.⌋

[constr_1448] Interval of `DiagnosticParameterIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈The allowed interval for values of `DiagnosticParameterIdentifier.id` shall not exceed [0..255].⌋

[constr_1449] PID shall only carry a fixed-length collection of data

Imposition time: CP: `IT_DiagDes`

⌈The value of `DiagnosticParameterIdentifier.dataElement.dataElement.arraySizeSemantics` shall not be set to `variableSize`.⌋

[constr_1450] Service mapping for OBD mode 0x01 for `DiagnosticParameterIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` refers to a `DiagnosticRequestCurrentPowertrainData` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticParameterIdentifier` then one of two alternative model configurations shall exist:

- `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.
- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.

⌋

[constr_1451] Service mapping for OBD mode 0x09 for `DiagnosticInfoType`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceSwMapping` refers to a `DiagnosticRequestVehicleInfo` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticInfoType` then one of two alternative model configurations shall exist:

- The `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` shall aggregate a `ObdInfoServiceNeeds` in the role `serviceNeeds`.
- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdInfoServiceNeeds` in the role `serviceNeeds`.

⌋

[constr_1452] Service mapping for OBD mode 0x08 for DiagnosticInfoType

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticServiceSwMapping refers to a DiagnosticRequestControlOfOnBoardDevice then the SwcServiceDependency referenced by the same DiagnosticServiceSwMapping shall aggregate an ObdControlServiceNeeds in the role serviceNeeds.⌋

[constr_1453] References from DiagnosticFunctionInhibitSource

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticFunctionInhibitSource may either reference one of the following meta-classes in the respective role:

- DiagnosticFimAliasEventMapping in the role event
- DiagnosticFimAliasEventGroupMapping in the role eventGroup

⌋

[constr_1454] DiagnosticFimFunctionMapping shall only reference a SwcServiceDependency that aggregates FunctionInhibitionNeeds

Imposition time: CP: IT_DiagDes

⌈A DiagnosticFimFunctionMapping shall only reference a SwcServiceDependency that aggregates FunctionInhibitionNeeds in the role serviceNeeds.⌋

[constr_1455] Relation of DiagnosticJ1939Node to J1939NmNode

Imposition time: CP: IT_DiagDes

⌈Each J1939NmNode shall only be referenced in the role nmNode by a single DiagnosticJ1939Node.⌋

[constr_1456] Valid interval for attribute DiagnosticTroubleCodeJ1939.fmi

Imposition time: CP: IT_DiagDes

⌈The value of the attribute DiagnosticTroubleCodeJ1939.fmi shall be in the interval 0..31.⌋

[constr_1457] Service-only DTCs shall refer to a common memory section

Imposition time: CP: IT_DiagDes

⌈All DiagnosticTroubleCodeJ1939 with attribute kind set to the value serviceOnly that reference the same DiagnosticJ1939Node shall also reference the same DiagnosticTroubleCodeProps.diagnosticMemory.⌋

[constr_1458] Reference to DiagnosticMemoryDestination

Imposition time: CP: IT_DiagDes

⌈A DiagnosticMemoryDestination that is referenced by a DiagnosticTroubleCodeJ1939.dtcProps.diagnosticMemory where the value of attribute

DiagnosticTroubleCodeJ1939.kind is set to serviceOnly shall **not be referenced by any other** DiagnosticTroubleCodeJ1939 where attribute kind is set to any other value than serviceOnly.⌋

[constr_1460] Restrictions for the value of DiagnosticInfoType.id

Imposition time: CP: IT_DiagDes

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticInfoType.id.⌋

[constr_1461] Restrictions for the value of DiagnosticTestRoutineIdentifier.id

Imposition time: CP: IT_DiagDes

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticTestRoutineIdentifier.id.⌋

[constr_1462] Restrictions for the value of DiagnosticTestResult.testIdentifier.id

Imposition time: CP: IT_DiagDes

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticTestResult.testIdentifier.id.⌋

[constr_1464] Allowed value range of DiagnosticEnvConditionFormula.nrcValue

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticEnvConditionFormula.nrcValue shall be limited to the interval [1..255].⌋

[constr_1466] Allowed values of compareType in the context of a DiagnosticEnvModeCondition

Imposition time: CP: IT_DiagDes

⌈Within the context of a DiagnosticEnvDataCondition **only a subset** of the values of DiagnosticCompareTypeEnum is supported for the inherited attribute compareType, namely:

- DiagnosticCompareTypeEnum.isEqual
- DiagnosticCompareTypeEnum.isNotEqual

⌋

[constr_1467] References in DiagnosticEnvModeCondition

Imposition time: CP: IT_DiagDes

⌈In a DiagnosticEnvModeCondition the reference modeElement shall only point to a DiagnosticEnvModeElement that is aggregated inside the same

DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition itself.⌋

[constr_1470] Value of DiagnosticAbstractParameter.bitOffset

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of DiagnosticAbstractParameter.bitOffset shall only be set to a multiple of 8.⌋

[constr_1472] Existence of DiagnosticDataIdentifier.supportInfoByte

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The attribute DiagnosticDataIdentifier.supportInfoByte **shall not exist** if the value of DiagnosticDataIdentifier.id is **outside the range 0xF400-0xF4FF**.⌋

[constr_1509] Value of extendedDataRecord.recordNumber shall be unique within primary fault memory

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationPrimary in the role diagnosticMemory there shall be no two extendedDataRecord.recordNumber with the same value.⌋

[constr_1511] Value of extendedDataRecord.recordNumber shall be unique within user-defined fault memory

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationUserDefined in the role diagnosticMemory there shall be no two extendedDataRecord.recordNumber with the same value for any DiagnosticMemoryDestinationUserDefined referenced as DiagnosticTroubleCodeProps.diagnosticMemory with a given value of memoryId.⌋

[constr_1512] freezeFrame.recordNumber shall be unique within primary fault memory

Imposition time: CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationPrimary in the role diagnosticMemory there shall be no two freezeFrame.recordNumber with the same value.⌋

[constr_1514] freezeFrame.recordNumber shall be unique within user-defined fault memory

Imposition time: CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationUserDefined in the role diagnosticMemory there shall be no two freezeFrame.recordNumber with the same value for any Diagnostic-

MemoryDestinationUserDefined referenced as DiagnosticTroubleCodeProps.diagnosticMemory with a given value of memoryId.⌋

[constr_1552] Restrictions for DiagnosticDataIdentifierSet referencing meta-class DiagnosticDataIdentifier

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticDataIdentifier is referenced by a DiagnosticDataIdentifierSet, then the DiagnosticDataIdentifier shall fulfill **all of the following conditions**:

- The DiagnosticDataIdentifier does **not** aggregate any DiagnosticParameter which in turn aggregates (via the aggregation of DiagnosticParameterIdent) a DiagnosticParameterElement.
- Gaps in between individual elements (i.e. under consideration of the beginning of an individual DiagnosticDataElement, as indicated by DiagnosticParameter.bitOffset, and the length of the aggregated DiagnosticDataElement) or at the end of the DiagnosticDataIdentifier (as indicated by attribute DiagnosticDataIdentifier.didSize) shall **not** exist.

The individual DiagnosticDataElement contained (via the aggregation of DiagnosticParameter in the role dataElement) in the DiagnosticDataIdentifier shall satisfy **one of the following conditions**:

- The DiagnosticDataElement does **not** define the attribute maxNumberOfElements at all.
- The modeling of attribute DiagnosticDataElement.arraySizeSemantics shall follow the description in [TPS_DEXT_01001].

⌋

[constr_1584] DiagnosticDataElement shall not be used more than once in I/O Control instance

Imposition time: CP: IT_DiagDes

⌈A given DiagnosticDataElement shall not be used by more than one DiagnosticServiceDataMapping that in turn refers to a DataPrototype defined in the context of a DataInterface that is used to type a PortPrototype that in turn is referenced by a RoleBasedPortAssignment where attribute role is set to the value IOControlRequest.⌋

[constr_1590] DiagnosticEvent referenced in the role masterEvent or slaveEvent

Imposition time: CP: IT_DiagDes

⌈Any given DiagnosticEvent shall be referenced at most once from a DiagnosticMasterToSlaveEventMapping.⌋

[constr_1591] `DiagnosticEvent` referenced as `slaveEvent` shall not be reported by diagnostic monitor

Imposition time: CP: `IT_DiagDes`

⌈Any `DiagnosticEvent` that is referenced in the role `DiagnosticMasterToSlaveEventMapping.slaveEvent` shall not be referenced in the role `DiagnosticEventPortMapping.diagnosticEvent` and vice versa.⌋

[constr_1616] Existence of attribute `DiagnosticExtendedDataRecord.customTrigger`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈The attribute `DiagnosticExtendedDataRecord.customTrigger` shall only exist if the attribute `DiagnosticExtendedDataRecord.trigger` is set to the value `DiagnosticRecordTriggerEnum.custom`.⌋

[constr_1617] Existence of attribute `DiagnosticFreezeFrame.customTrigger`

Imposition time: CP: `IT_DiagDes`

⌈The attribute `DiagnosticFreezeFrame.customTrigger` shall only exist if the attribute `DiagnosticFreezeFrame.trigger` is set to the value `DiagnosticRecordTriggerEnum.custom`.⌋

[constr_1623] Restriction on `DiagnosticReadScalingDataByIdentifier.dataIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈The reference `DiagnosticReadScalingDataByIdentifier.dataIdentifier` shall only refer to a `DiagnosticDataIdentifier`.⌋

[constr_1624] Existence of `DiagnosticDataElement.scalingInfoSize`

Imposition time: CP: `IT_DiagDes`

⌈The attribute `DiagnosticDataElement.scalingInfoSize` shall only exist if the enclosing `DiagnosticAbstractParameter` is aggregated by a `DiagnosticDataIdentifier` that is referenced by a `DiagnosticReadScalingDataByIdentifier` in the role `DiagnosticReadScalingDataByIdentifier.dataIdentifier`.⌋

[constr_1721] `DiagnosticControlEnableMaskBit.bitNumber` shall be unique

Imposition time: CP: `IT_DiagDes`

⌈Within the context of the enclosing `DiagnosticIOControl` the value of attribute `DiagnosticIOControl.controlEnableMaskBit.bitNumber` shall be unique.⌋

[constr_1722] Relation between reference DiagnosticIOControl.dataIdentifier and attribute DiagnosticIOControl.controlEnableMaskBit

Imposition time: CP: IT_DiagDes

⌈Any DiagnosticDataElement referenced in the role DiagnosticIOControl.controlEnableMaskBit.controlledDataElement shall be defined in the scope of the DiagnosticDataIdentifier that is referenced in the role DiagnosticIOControl.dataIdentifier.⌋

[constr_1745] Indirect reference to DiagnosticCommonElement

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If a DiagnosticCommonElement is referenced from within the context of another DiagnosticCommonElement and the referencing DiagnosticCommonElement is in turn referenced by a DiagnosticContributionSet in the role element then the referenced DiagnosticCommonElement shall also be referenced by the same DiagnosticContributionSet.⌋

[constr_1749] Existence of DiagnosticInfoType.dataElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticInfoType, at least one aggregation of DiagnosticParameter in the role dataElement shall exist.⌋

[constr_1750] Existence of attribute DiagnosticParameterIdentifier.pidSize

Imposition time: CP: IT_DiagDes

⌈Attribute DiagnosticParameterIdentifier.pidSize is only relevant if a gap exists at the end of the DiagnosticParameterIdentifier. If this gap does not exist, the size of the DiagnosticParameterIdentifier can be computed.⌋

[constr_1752] Existence of references owned by meta-class DiagnosticEnableConditionPortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnableConditionPortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

may exist.⌋

[constr_1753] Existence of references owned by meta-class DiagnosticStorageConditionPortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticStorageConditionPortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

may exist.⌋

[constr_1756] Existence of attributes DiagnosticExtendedDataRecord.trigger and update

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticExtendedDataRecord, attributes trigger and update shall only exist if at least one DiagnosticExtendedDataRecordElement is aggregated by a DiagnosticExtendedDataRecord in the role element that is referenced in either of the roles

- DiagnosticEdrSenderPortMapping.recordElement
- DiagnosticEdrServerPortMapping.recordElement

⌋

[constr_1757] Existence of the attribute DiagnosticTroubleCodeUds.udsDtcValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUds, attribute udsDtcValue shall exist.⌋

[constr_1758] Existence of primitive attribute DiagnosticTroubleCodeObd.obdDTCValue

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeObd, attribute obdDTCValue shall exist.⌋

[constr_1759] Existence of references owned by DiagnosticOperationCyclePortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticOperationCyclePortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

shall exist.⌋

[constr_1760] Existence of DiagnosticExtendedDataRecord.element

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticExtendedDataRecord, at least one aggregation of DiagnosticExtendedDataRecordElement in the role element shall exist.⌋

[constr_1762] Existence of references owned by DiagnosticEventPortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventPortMapping, only one of the references

- to BswServiceDependency in the role bswServiceDependency
- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

shall exist.⌋

[constr_1763] Existence of attribute DiagnosticPeriodicRate.periodicRateCategory

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticPeriodicRate, the attribute periodicRateCategory shall exist.⌋

[constr_1766] Existence of the attribute DiagEventDebounceCounterBased.counterJumpDownValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterJumpDownValue shall only exist if attribute counterJumpDown exists and is set to true.⌋

[constr_1767] Existence of the attribute DiagEventDebounceCounterBased.counterJumpUpValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterJumpUpValue shall only exist if attribute counterJumpUp exists and is set to true.⌋

[constr_1768] Existence of attribute DiagnosticEvent.associatedEventIdentification

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Attribute DiagnosticEvent.associatedEventIdentification shall exist if

- the respective DiagnosticEvent is mapped to a DiagnosticTroubleCodeUds and
- the reference in the role DiagnosticTroubleCodeUds.troubleCodeProps.extendedDataRecord exists and
- the referenced DiagnosticExtendedDataRecord aggregates at least one element that is also referenced by a DiagnosticEdrDataProviderMapping that has attribute dataProvider set to the value DEM_EVENT_ASSOCIATED_IDENTIFICATION.

⌋

[constr_1772] Unique DiagnosticSession and DiagnosticSecurityLevel for diagnostic routines that have the same identifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈All DiagnosticAccessPermissions referenced from DiagnosticRoutines where attribute DiagnosticRoutine.id has the identical value shall refer to the identical set of DiagnosticSession and DiagnosticSecurityLevel.⌋

[constr_1780] Existence of attribute DiagnosticTroubleCodeJ1939.fmi

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeJ1939, attribute fmi shall exist.⌋

[constr_1781] Existence of attribute DiagnosticTroubleCodeJ1939.spn

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeJ1939, attribute spn shall exist.⌋

[constr_1790] Existence of attribute DiagnosticAbstractParameter.bitOffset

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticParameter, attribute bitOffset shall exist.⌋

[constr_1791] Existence of attribute dataElement vs. parameterSize of meta-class DiagnosticParameter

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticParameter, exactly **one** of the attributes

- dataElement or
- parameterSize

shall exist.⌋

[constr_1792] Existence of DiagnosticDataIdentifier.dataElement

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticDataIdentifier, the aggregation of DiagnosticParameter in the role dataElement shall exist at least once.⌋

[constr_1793] Existence of attribute DiagnosticAbstractDataIdentifier.id

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticAbstractDataIdentifier, attribute id shall exist.⌋

[constr_1794] Existence of attribute DiagnosticProtocol.priority

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticProtocol, attribute priority shall exist.⌋

[constr_1795] Existence of attribute DiagnosticProtocol.protocolKind

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticProtocol, attribute protocolKind shall exist.⌋

[constr_1797] Existence of attribute DiagnosticServiceTable.protocolKind

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticServiceTable, attribute protocolKind shall exist.⌋

[constr_1798] Existence of DiagnosticServiceInstance.serviceClass

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each subclass of DiagnosticServiceInstance, a reference with the abstract role serviceClass shall exist to a matching subclass of DiagnosticServiceClass.

This rule applies unless a rule for a specific combination of matching sub-classes of DiagnosticServiceInstance and DiagnosticServiceClass exists.⌋

[constr_1799] Existence of DiagnosticEnvironmentalCondition.formula

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvironmentalCondition, the aggregation of DiagnosticEnvConditionFormula in the role formula shall exist.⌋

[constr_1800] Existence of DiagnosticEnvConditionFormula.op

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvConditionFormula, the attribute op shall exist.⌋

[constr_1801] Existence of DiagnosticEnvCompareCondition.compareType

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvCompareCondition, that attribute compareType shall exist.⌋

[constr_1802] Existence of DiagnosticEnvDataCondition.compareValue

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvDataCondition, that attribute compareValue shall exist.⌋

[constr_1803] Existence of DiagnosticEnvDataCondition.dataElement

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvDataCondition, that attribute dataElement shall exist.⌋

[constr_1804] Existence of DiagnosticEnvModeCondition.modeElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnvModeCondition, that attribute modeElement shall exist.⌋

[constr_1805] Existence of DiagnosticEnvSwcModeElement.mode

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnvSwcModeElement, that attribute mode shall exist.⌋

[constr_1806] Existence of DiagnosticEnvBswModeElement.mode

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnvBswModeElement, that attribute mode shall exist.⌋

[constr_1807] Existence of reference DiagnosticDataByIdentifier.dataIdentifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticDataByIdentifier, the reference dataIdentifier shall exist.⌋

[constr_1808] Existence of reference DiagnosticDynamicallyDefineDataIdentifier.dataIdentifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticDynamicallyDefineDataIdentifier, the reference to DiagnosticDynamicDataIdentifier in the role dataIdentifier shall exist.⌋

[constr_1810] Existence of the aggregation DiagnosticReadDataByPeriodicIDClass.periodicRate

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the aggregation of DiagnosticPeriodicRate in the role periodicRate shall exist at least once.⌋

[constr_1811] Existence of the attribute DiagnosticReadDataByPeriodicIDClass.maxPeriodicDidToRead

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the attribute maxPeriodicDidToRead shall exist at least once.⌋

[constr_1812] Existence of the attribute DiagnosticReadDataByPeriodicIDClass.schedulerMaxNumber

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the attribute schedulerMaxNumber shall exist at least once.⌋

[constr_1815] Existence of attribute DiagnosticRoutine.id

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticRoutine, the attribute id shall exist at least once.⌋

[constr_1816] Existence of attribute DiagnosticSecurityAccess.requestSeedId

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSecurityAccess, the attribute requestSeedId shall exist at least once.⌋

[constr_1817] Existence of attribute DiagnosticSecurityAccess.securityLevel

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSecurityAccess, the attribute securityLevel shall exist at least once.⌋

[constr_1818] Existence of reference in the role DiagnosticSessionControl.diagnosticSession

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSessionControl, the reference to DiagnosticSession in the role diagnosticSession shall exist.⌋

[constr_1819] Existence of attribute DiagnosticParameterIdentifier.id

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticParameterIdentifier, attribute id shall exist.⌋

[constr_1820] Existence of reference in the role DiagnosticRequestCurrentPowertrainData.pid

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticRequestCurrentPowertrainData, the reference to meta-class DiagnosticParameterIdentifier in the role pid shall exist.⌋

[constr_1821] Existence of reference DiagnosticRequestPowertrainFreezeFrameData.freezeFrame

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticRequestPowertrainFreezeFrameData, the reference to DiagnosticParameterIdentifier in the role freezeFrame shall exist.⌋

[constr_1822] Existence of reference DiagnosticRequestControlOfOnBoardDevice.testId

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticRequestControlOfOnBoardDevice, the reference to DiagnosticParameterIdentifier in the role testId shall exist.⌋

[constr_1823] Existence of attribute `DiagnosticTestRoutineIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `id` shall exist.⌋

[constr_1824] Existence of attribute `DiagnosticTestRoutineIdentifier.requestDataSize`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `requestDataSize` shall exist.⌋

[constr_1825] Existence of attribute `DiagnosticTestRoutineIdentifier.responseDataSize`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `responseDataSize` shall exist.⌋

[constr_1826] Existence of reference `DiagnosticRequestVehicleInfo.infoType`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticRequestVehicleInfo`, the reference to `DiagnosticParameterIdentifier` in the role `infoType` shall exist.⌋

[constr_1827] Existence of attribute `DiagnosticInfoType.id`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticInfoType`, attribute `id` shall exist.⌋

[constr_1828] Existence of references from `DiagnosticServiceDataMapping`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticServiceDataMapping`, the following references shall exist:

- Reference to a `DiagnosticServiceMappingDiagTarget`, i.e. one of
  - Reference to `DiagnosticDataElement` in the role `diagnosticDataElement`
  - Reference to `DiagnosticParameterIdent` in the role `diagnosticParameter`
  - Reference to `DiagnosticParameterElement` in the roles
    - `contextElement` (optional)
    - `targetElement`

    from within `parameterElementAccess`
- Reference to `DataPrototype` in the role `mappedDataElement`

⌋

[constr_1829] Existence of reference DiagnosticConnectedIndicator.indicator

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticConnectedIndicator, the reference to DiagnosticIndicator in the role indicator shall exist.⌋

[constr_1830] Existence of DiagnosticTroubleCodeGroup.groupNumber

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeGroup, attribute groupNumber shall exist.⌋

[constr_1831] Existence of DiagnosticTroubleCodeProps.priority

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeProps, attribute priority shall exist.⌋

[constr_1832] Existence of DiagnosticExtendedDataRecord.recordNumber

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticExtendedDataRecord, attribute recordNumber shall exist.⌋

[constr_1833] Existence of DiagnosticFreezeFrame.trigger

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticFreezeFrame, attribute trigger shall exist.⌋

[constr_1834] Existence of DiagnosticCondition.initValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticCondition, attribute initValue shall exist.⌋

[constr_1835] Existence of the attribute DiagEventDebounceCounterBased.counterDecrementStepSize

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterDecrementStepSize shall exist.⌋

[constr_1836] Existence of the attribute DiagEventDebounceCounterBased.counterIncrementStepSize

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterIncrementStepSize shall exist.⌋

[constr_1837] Existence of the attribute DiagEventDebounceCounterBased.counterFailedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterFailedThreshold shall exist.⌋

[constr_1838] Existence of the attribute DiagEventDebounceCounterBased.counterPassedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterPassedThreshold shall exist.⌋

[constr_1839] Existence of the attribute DiagEventDebounceTimeBased.timeFailedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceTimeBased, attribute timeFailedThreshold shall exist.⌋

[constr_1840] Existence of attribute DiagEventDebounceTimeBased.timePassedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceTimeBased, attribute timePassedThreshold shall exist.⌋

[constr_1841] Existence of attribute DiagnosticEnableConditionGroup.enableCondition

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticEnableConditionGroup, the attribute enableCondition shall exist.⌋

[constr_1842] Existence of DiagnosticStorageConditionGroup.storageCondition

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticStorageConditionGroup, attribute storageCondition shall exist.⌋

[constr_1843] Existence of reference DiagnosticEventPortMapping.diagnosticEvent

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventPortMapping, the reference to DiagnosticEvent in the role diagnosticEvent shall exist.⌋

[constr_1844] Existence of reference DiagnosticOperationCyclePortMapping.operationCycle

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticOperationCyclePortMapping, the reference to DiagnosticOperationCycle in the role operationCycle shall exist.⌋

[constr_1845] Existence of reference DiagnosticEnableConditionPortMapping.enableCondition

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnableConditionPortMapping, the reference to DiagnosticEnableCondition in the role enableCondition shall exist.⌋

[constr_1846] Existence of reference DiagnosticStorageConditionPortMapping.diagnosticStorageCondition

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticStorageConditionPortMapping, the reference to DiagnosticStorageCondition in the role diagnosticStorageCondition shall exist.⌋

[constr_1847] Existence of reference DiagnosticEdrDataProviderMapping.extendedDataRecordElement

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticEdrDataProviderMapping, the reference to DiagnosticExtendedDataRecordElement in the role extendedDataRecordElement shall exist.⌋

[constr_1848] Existence of attribute DiagnosticAging.agingCycle

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticAging, attribute agingCycle shall exist.⌋

[constr_1849] Existence of attribute DiagnosticAging.threshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticAging, attribute threshold shall exist.⌋

[constr_1850] Existence of the aggregation in the role DiagnosticTestResult.testIdentifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTestResult, the aggregation of meta-class DiagnosticTestIdentifier in the role testIdentifier shall exist.⌋

[constr_1851] Existence of the reference DiagnosticTestResult.monitoredIdentifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTestResult, the reference to meta-class DiagnosticTestIdentifier in the role monitoredIdentifier shall exist.⌋

[constr_1852] Existence of attribute DiagnosticEcuInstanceProps.obdSupport

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEcuInstanceProps, attribute obdSupport shall exist.⌋

[constr_1853] Existence of attribute DiagnosticIumprGroup.iumprGroupIdentifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticIumprGroup, attribute iumprGroupIdentifier shall exist.⌋

[constr_1854] Existence of attribute DiagnosticIumprGroupIdentifier.groupId

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticIumprGroupIdentifier, attribute groupId shall exist.⌋

[constr_1855] Existence of the attribute DiagnosticFunctionIdentifierInhibit.inhibitionMask

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticFunctionIdentifierInhibit, attribute inhibitionMask shall exist.⌋

[constr_1856] Existence of attribute DiagnosticJ1939Spn.spn

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticJ1939Spn, attribute spn shall exist.⌋

[constr_1857] Existence of the reference DiagnosticEventToTroubleCodeJ1939Mapping.diagnosticEvent

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventToTroubleCodeJ1939Mapping, the reference in the role diagnosticEvent shall exist.⌋

[constr_1858] Existence of the attribute DiagnosticEventToTroubleCodeJ1939Mapping.troubleCodeJ1939

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventToTroubleCodeJ1939Mapping, attribute troubleCodeJ1939 shall exist.⌋

[constr_1859] Usage of DiagnosticRecordTriggerEnum.testFailedThis-OperationCycle

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The enumeration value DiagnosticRecordTriggerEnum.testFailedThis-OperationCycle shall only be used in the context of meta-class Diagnostic-FreezeFrame.⌋

[constr_10026] Existence of the reference in the role DiagnosticEventTo-SecurityEventMapping.diagnosticEvent

Status: DRAFT
Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEventToSecurityEventMapping, the reference to DiagnosticEvent in the role diagnosticEvent shall exist.⌋

[constr_10027] Existence of the reference in the role DiagnosticEventTo-SecurityEventMapping.securityEventProps

Status: DRAFT
Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEventToSecurityEventMapping, the reference to SecurityEventContextProps in the role securityEventProps shall exist.⌋

[constr_10038] Restriction for the usage of DiagnosticAccessPermission.authenticationEnabled

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticAccessPermission.authenticationEnabled shall not exist if the DiagnosticAccessPermission is referenced from

- DiagnosticRequestCurrentPowertrainData
- DiagnosticRequestPowertrainFreezeFrameData
- DiagnosticRequestEmissionRelatedDTC
- DiagnosticClearResetEmissionRelatedInfo
- DiagnosticRequestOnBoardMonitoringTestResults
- DiagnosticRequestControlOfOnBoardDevice
- DiagnosticRequestVehicleInfo
- DiagnosticRequestEmissionRelatedDTCPermanentStatus
- sub-classes of DiagnosticAuthentication

⌋

[constr_10042] Existence of the attribute DiagnosticCommonProps.defaultEndianness

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈One of the following conditions shall be fulfilled:

- DiagnosticCommonProps.defaultEndianness exists.
- The primitive attribute DiagnosticParameter.dataElement.swDataDefProps.baseType.baseTypeDefinition.baseTypeEncoding exists for **all** DiagnosticParameters defined in the context of the DiagnosticContributionSet.

⌋

[constr_10043] Existence of the primitive attribute DiagnosticCommonProps.resetConfirmedBitOnOverflow

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticCommonProps.resetConfirmedBitOnOverflow shall exist.⌋

[constr_10044] Existence of primitive attribute DiagnosticCommonProps.occurrenceCounterProcessing

Imposition time: CP: IT_DiagDes

⌈If, in the context of a DiagnosticContributionSet, a DiagnosticEdrDataProviderMapping exists where attribute DiagnosticEdrDataProviderMapping.dataProvider is set to the value DEM_OCCCTR, then attribute DiagnosticCommonProps.occurrenceCounterProcessing shall exist.⌋

[constr_10045] Existence of attribute DiagnosticSecurityAccess.securityDelayTimeOnBoot

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticSecurityAccess.securityDelayTimeOnBoot shall exist.⌋

[constr_10084] Existence of role DiagnosticIumprToFunctionIdentifierMapping.iumpr

Imposition time: CP: IT_DiagDes

⌈For all DiagnosticIumprToFunctionIdentifierMapping, the reference in the role iumpr shall exist.⌋

[constr_10085] Existence of role DiagnosticIumprToFunctionIdentifierMapping.functionIdentifier

Imposition time: CP: IT_DiagDes

⌈For all DiagnosticIumprToFunctionIdentifierMapping, the reference in the role functionIdentifier shall exist.⌋

[constr_10088] Relation between event and DTC without event combination

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If the primitive attribute DiagnosticCommonProps.typeOfEventCombinationSupported is not configured, then all DiagnosticTroubleCodeUds that refer to a DiagnosticTroubleCodeProps in the role troubleCodeProps that in turn refers to a DiagnosticMemoryDestination in the role diagnosticMemory shall only be referenced by at most one DiagnosticEventToTroubleCodeUdsMapping.⌋

[constr_10089] Existence of the attribute DiagnosticCommonProps.eventCombinationReportingBehavior

Imposition time: CP: IT_DiagDes

⌈Attribute DiagnosticCommonProps.eventCombinationReportingBehavior is always optional and shall be set to the value DiagnosticEventCombinationReportingBehaviorEnum.reportingInChronologicalOrderOldestFirst only if the attribute DiagnosticCommonProps.typeOfEventCombinationSupported is set to the value DiagnosticEventCombinationBehaviorEnum.eventCombinationOnRetrieval.

If it is missing, then the reporting order is not specified.⌋

[constr_10091] Mandatory subfunction of diagnostic service Authentication

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If the diagnostic service Authentication is supported, then the following subfunctions shall be configured:

- De-authentication, formalized by meta-class DiagnosticDeAuthentication.
- Proof of ownership, formalized by meta-class DiagnosticProofOfOwnership.
- Authentication configuration, formalized by meta-class DiagnosticAuthenticationConfiguration.
- One of
  - Verify certificate unidirectional, formalized by meta-class DiagnosticVerifyCertificateUnidirectional.
  - Verify certificate bidirectional, formalized by meta-class DiagnosticVerifyCertificateBidirectional.

⌋

[constr_10100] Existence of DiagnosticRoutineControl.routine

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticRoutineControl, the attribute routine shall exist.⌋

[constr_10115] Existence of attributes of meta-class DiagnosticEnvDataElementCondition if the reference in the role dataPrototype exists

Imposition time: CP: IT_DiagDes

⌈If the reference in the role DiagnosticEnvDataElementCondition.dataPrototype exists, then

- the aggregation in the role compareValue shall exist and
- the aggregation in the role swDataDefProps shall not exist.

⌋

[constr_10116] Existence of attributes of meta-class DiagnosticEnvDataElementCondition if the reference in the role dataPrototype does not exist

Imposition time: CP: IT_DiagDes

⌈If the reference in the role DiagnosticEnvDataElementCondition.dataPrototype does **not** exist, then the aggregations in the role

- compareValue and
- swDataDefProps

shall exist.⌋

[constr_10117] Existence of attributes of DiagnosticEnvDataElementCondition.swDataDefProps

Imposition time: CP: IT_DiagDes

⌈

Attribute of SwDataDefProps	Attribute Existence
baseType	1
compuMethod	0..1
dataConstr	0..1

⌋

[constr_10122] Existence of the attribute DiagnosticComControlSubNodeChannel.subNodeChannel

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticComControlSubNodeChannel.subNodeChannel shall only exist if the value of DiagnosticComControl.category is set to either

- ENABLE_RX_AND_DISABLE_TX_WITH_ENHANCED_ADDRESS_INFORMATION or
- ENABLE_RX_AND_TX_WITH_ENHANCED_ADDRESS_INFORMATION.

⌋

[constr_10364] Usage of DiagnosticRecordTriggerEnum.testPassed

Imposition time: CP: IT_DiagDes

⌈The enumeration value DiagnosticRecordTriggerEnum.testPassed shall only be used in the context of DiagnosticExtendedDataRecord.trigger.⌋

[constr_10368] Restriction regarding the reference in the role DiagnosticDataIdentifierSet.diagnosticDataIdentifier

Imposition time: CP: IT_DiagDes

⌈A DiagnosticDataIdentifier that is referenced in the role DiagnosticDataIdentifierSet.diagnosticDataIdentifier shall not aggregate in the role dataElement a DiagnosticParameter that aggregates in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.⌋

[constr_10369] Existence of attributes of DiagnosticParameterElement depending on the value of attribute category

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈

Value of category	Description	arraySize	subElement	dataElement
LEAF	The DiagnosticParameterElement represents a "leaf" element of a nested structure.	No	No	Yes
ARRAY	The DiagnosticParameterElement represents an array, i.e references to this DiagnosticParameterElement shall define a value for the attribute index. Arrays of "primitive" types are defined in the context of the DiagnosticDataElement	Yes	Yes, if dataElement does not exist	Yes, if subElement does not exist
STRUCTURE	The DiagnosticParameterElement represents a structure with one or more elements.	No	Yes	No

⌋

[constr_10370] Restriction regarding the role DiagnosticParameterIdentifier.dataElement

Imposition time: CP: IT_DiagDes

⌈A DiagnosticParameter that is aggregated by a DiagnosticParameterIdentifier in the role dataElement shall not aggregate in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.⌋

[constr_10412] Existence of attribute DiagnosticTestIdentifier.id

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTestIdentifier, attribute id shall exist.⌋

[constr_10413] Existence of attribute DiagnosticTestIdentifier.uasId

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTestIdentifier, attribute uasId shall exist.⌋

[constr_10414] Existence of attribute DiagnosticMeasurementIdentifier.obdMid

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticMeasurementIdentifier, attribute obdMid shall exist.⌋

[constr_10418] Existence of attribute DiagnosticDebounceAlgorithmProps.debounceAlgorithm

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticDebounceAlgorithmProps, the primitive attribute debounceAlgorithm shall exist.⌋

[constr_10419] Existence of the attribute DiagnosticCommonProps.resetPendingBitOnOverflow

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticCommonProps.resetPendingBitOnOverflow shall exist.⌋

[constr_10421] Existence of attribute DiagnosticMemoryDestination.dtcStatusAvailabilityMask

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, the primitive attribute dtcStatusAvailabilityMask shall exist.⌋

[constr_10422] Existence of attribute DiagnosticMemoryDestination.eventDisplacementStrategy

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, attribute eventDisplacementStrategy shall exist.⌋

[constr_10423] Existence of attribute DiagnosticMemoryDestination.maxNumberOfEventEntries

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, attribute maxNumberOfEventEntries shall exist.⌋

[constr_10440] Restriction for the minimum value of attribute DiagnosticSessionControlClass.s3ServerTimeout

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticSessionControlClass.s3ServerTimeout shall be **greater than or equal to** 5.0.⌋

[constr_10522] OBD trouble code shall only be placed in primary fault memory

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticTroubleCodeUds that is referenced in the role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds in turn references a DiagnosticTroubleCodeProps in the role troubleCodeProps, and if the DiagnosticTroubleCodeProps refers to a DiagnosticMemoryDestination in the role diagnosticMemory, then the referenced diagnosticMemory shall be a DiagnosticMemoryDestinationPrimary.⌋

[constr_10523] Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUdsToTroubleCodeObdMapping, the reference in the role troubleCodeUds shall exist.⌋

[constr_10524] Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeObd

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUdsToTroubleCodeObdMapping, the reference in the role troubleCodeObd shall exist.⌋

[constr_10545] Existence of DiagnosticParameterIdentifier.dataElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticParameterIdentifier, at least one aggregation of DiagnosticParameter in the role DiagnosticParameterIdentifier.dataElement shall exist.⌋

[constr_10573] Existence of attribute DiagnosticServiceTable.diagnosticServiceInstance

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticServiceTable, attribute diagnosticServiceInstance shall exist.⌋

[constr_10577] Existence of DiagnosticResponseOnEventClass.storeEventEnabled

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticResponseOnEventClass, the attribute storeEventEnabled shall exist.⌋

[constr_10642] Existence of DiagnosticEventPortMapping.diagnosticEvent for one specific DiagnosticEvent

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticEvent shall be referenced **at most once** in the role DiagnosticEventPortMapping.diagnosticEvent.⌋

[constr_10645] Value of DiagnosticEvent.confirmationThreshold for an n:1 mapping to DiagnosticTroubleCodeUds

Imposition time: CP: IT_DiagDes

⌈If multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) associated with that same DiagnosticTroubleCodeUds, then the value of attribute confirmationThreshold of **each** of the affected DiagnosticEvents shall be **identical**.⌋

[constr_10646] Value of attributes of DiagnosticConnectedIndicator for an n:1 mapping to DiagnosticTroubleCodeUds where all affected DiagnosticEvents transitively refer to the identical DiagnosticIndicator

Imposition time: CP: IT_DiagDes

⌈If

- multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) associated with the same DiagnosticTroubleCodeUds **and**
- the target of **all** connectedIndicator.indicator is the identical DiagnosticIndicator,

then the values of the following attributes and references owned by **all** affected DiagnosticConnectedIndicators shall have **the identical value**:

- behavior
- indicatorFailureCycleCounterThreshold
- healingCycleCounterThreshold
- indicator

⌋

[constr_10647] In an n:1 mapping between DiagnosticEvent and DiagnosticTroubleCodeUds, all DiagnosticEvents shall be mapped to the same DiagnosticOperationCycle

Imposition time: CP: IT_DiagDes

⌈If multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) mapped to the same DiagnosticTroubleCodeUds, then all affected DiagnosticEvents shall (by way of multiple Diagnostic-

EventToOperationCycleMappings) be mapped to the identical DiagnosticOperationCycle.⌋

[constr_10665] Existence of role DiagnosticCommonProps.authenticationTimeout

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If the DiagnosticContributionSet that aggregates a DiagnosticCommonProps also references a DiagnosticAuthentication in the role element, then attribute DiagnosticCommonProps.authenticationTimeout shall exist.⌋

[constr_10676] Allowed attributes of DiagnosticExtendedDataRecordElement.swDataDefProps

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The only allowed attributes of SwDataDefProps aggregated in the role DiagnosticExtendedDataRecordElement.swDataDefProps are

- compuMethod for the definition of a conversion method from internal to physical representation in the diagnostic client
- baseType for the definition of the memory layout of the DiagnosticExtendedDataRecord.

None of the other attributes of SwDataDefProps shall be used in the context of DiagnosticExtendedDataRecordElement.swDataDefProps.⌋

[constr_10677] Allowed values for attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The only allowed values filled into attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize shall be a multiple of 8.⌋

[constr_10678] Existence of the role DiagnosticEdrSenderPortMapping.dataPrototype

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrSenderPortMapping, the reference in the role DiagnosticEdrSenderPortMapping.dataPrototype shall exist.⌋

[constr_10679] Existence of DiagnosticEdrSenderPortMapping.recordElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrSenderPortMapping, the reference in the role DiagnosticEdrSenderPortMapping.recordElement shall exist.⌋

[constr_10680] Existence of role `DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem` vs. `bswServiceDependency`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEdrServerPortMapping`, **either** the reference in the role

- `DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem` **or**
- `DiagnosticEdrServerPortMapping.bswServiceDependency`

shall exist.⌋

[constr_10681] Existence of the role `DiagnosticEdrServerPortMapping.recordElement`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEdrServerPortMapping`, the reference in the role `DiagnosticEdrServerPortMapping.recordElement` shall exist.⌋

[constr_10687] Existence of the reference `ISignalPortToDiagnosticEventMapping.iSignalTriggering`

Status: DRAFT
Imposition time: CP: `IT_DiagDes`

⌈For each `ISignalPortToDiagnosticEventMapping`, the reference to meta-class`ISignalTriggering` in the role `iSignalTriggering` shall exist.⌋

[constr_10688] Existence of the reference `ISignalPortToDiagnosticEventMapping.iSignalPort`

Status: DRAFT
Imposition time: CP: `IT_DiagDes`

⌈For each `ISignalPortToDiagnosticEventMapping`, the reference to meta-class`ISignalPort` in the role `iSignalPort` shall exist.⌋

[constr_10689] Restriction for the attribute `communicationDirection` of an `ISignalPort` that is referenced by `ISignalPortToDiagnosticEventMapping`

Status: DRAFT
Imposition time: CP: `IT_DiagDes`

⌈Each `ISignalPort` that is referenced in the role `iSignalPort` by a `ISignalPortToDiagnosticEventMapping` shall set the attribute `communicationDirection` to the value `CommunicationDirectionType.in`.⌋

[constr_10750] Existence of `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping`

Imposition time: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeObd`, a `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping` shall exist that references the `DiagnosticTroubleCodeObd` in the role `troubleCodeObd`.⌋

[constr_10751] Existence of reference to `EventObdReadinessGroup`

Imposition time: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping`, **only** the referenced `troubleCodeObd` shall aggregate a `EventObdReadinessGroup` in the role `eventReadinessGroup` while the `troubleCodeUds` shall not aggregate a `EventObdReadinessGroup` in the role `eventReadinessGroup`.⌋

[constr_10752] Existence of `DiagnosticReadDataByPeriodicID` ⌈In the context of any `DiagnosticContributionSet`, at most one `DiagnosticReadDataByPeriodicID` can be defined.⌋

2.3 CP_TPS_ECUConfiguration

[constr_3022] `EcucModuleDef` category restriction

Imposition time: CP: `IT_EcucDef`

⌈The category definition shall be restricted to exactly the two defined ones:

- VENDOR_SPECIFIC_MODULE_DEFINITION
- STANDARDIZED_MODULE_DEFINITION

⌋

[constr_3023] Usage of `apiServicePrefix`

Imposition time: CP: `IT_EcucVal`

⌈The attribute `apiServicePrefix` is mandatory for VSMDs derived from the CDD and Xfrm StMD. The attribute shall not be provided for VSMDs derived from any other StMDs.⌋

[constr_3120] `EcucDestinationUriPolicy.container` always required

Imposition time: CP: `IT_EcucDef`

⌈`EcucDestinationUriPolicy.container` shall exist exactly once.⌋

[constr_3200] Restriction on values of `EcucDefinitionElement.relatedTraceItem` in the VSMD

Imposition time: CP: `IT_EcucDef`

⌈The value of `EcucDefinitionElement.relatedTraceItem` in the VSMD shall never start with 'ECUC_'.⌋

[constr_3217] Symbolic name reference shall point only to containers with a symbolic name value defined

Imposition time: CP: IT_EcucVal

⌈If an EcucReferenceValue exists that refers in the role definition to an EcucAbstractInternalReferenceDef with the attribute requiresSymbolicNameValue set to true, then the EcucContainerValue that is the target of the reference shall refer to an EcucParamConfContainerDef in the role definition that contains a definition of an EcucParameterDef where the attribute symbolicNameValue exists and is set to true. The EcucContainerValue shall define an EcucParameterValue that refers to an EcucParameterDef where the attribute symbolicNameValue exists and is set to true.⌋

[constr_3307] ShortNames of PredefinedVariants referenced by EcucPostBuildVariantRefs

Imposition time: CP: IT_EcucVal

⌈All PredefinedVariants that are referenced by EcucPostBuildVariantRefs shall have different shortNames.⌋

[constr_3450] postBuildVariantUsed value in case of post build VariationPoints

Imposition time: CP: IT_EcucVal

⌈If the configuration values of a BSW module contain at least one post build VariationPoint, the value of the postBuildVariantUsed for the EcucModuleConfigurationValues shall be set to TRUE.⌋

[constr_3451] No EcucModuleConfigurationValues.postBuildVariantUsed value setting restriction in case postBuildVariantSupport is set to TRUE

Imposition time: CP: IT_EcucVal

⌈If EcucModuleDef.postBuildVariantSupport is set to TRUE, then EcucModuleConfigurationValues.postBuildVariantUsed can be either TRUE or FALSE.⌋

[constr_3452] EcucModuleConfigurationValues.postBuildVariantUsed value setting restriction in case postBuildVariantSupport is set to FALSE

Imposition time: CP: IT_EcucVal

⌈If EcucModuleDef.postBuildVariantSupport is set to FALSE, then EcucModuleConfigurationValues.postBuildVariantUsed shall be FALSE.⌋

[constr_3570] EcucDefinitionElement.lowerMultiplicity always required

Imposition time: CP: IT_EcucDef

⌈The attribute EcucDefinitionElement.lowerMultiplicity shall always be defined.⌋

[constr_3571] `EcucCommonAttributes.origin` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucCommonAttributes.origin` shall always be defined.⌋

[constr_3572] `EcucParameterDef.symbolicNameValue` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucParameterDef.symbolicNameValue` shall always be defined.⌋

[constr_3575] `EcucEnumerationLiteralDef.origin` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucEnumerationLiteralDef.origin` shall always be defined.⌋

[constr_3576] `EcucInstanceReferenceDef.destinationContext` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucInstanceReferenceDef.destinationContext` shall always be defined.⌋

[constr_3577] `EcucInstanceReferenceDef.destinationType` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucInstanceReferenceDef.destinationType` shall always be defined.⌋

[constr_3578] `EcucForeignReferenceDef.destinationType` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucForeignReferenceDef.destinationType` shall always be defined.⌋

[constr_3579] `EcucReferenceDef.destination` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucReferenceDef.destination` shall always be defined.⌋

[constr_3580] `EcucUriReferenceDef.destinationUri` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucUriReferenceDef.destinationUri` shall always be defined.⌋

[constr_3581] `EcucDestinationUriDefSet.destinationUriDef` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucDestinationUriDefSet.destinationUriDef` shall always be defined.⌋

[constr_3582] `EcucDestinationUriDef.destinationUriPolicy` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucDestinationUriDef.destinationUriPolicy` shall always be defined.⌋

[constr_3584] `EcucQuery.ecucQueryExpression` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucQuery.ecucQueryExpression` shall always be defined.⌋

[constr_3585] `EcucConditionFormula.ecucQuery` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucConditionFormula.ecucQuery` shall always be defined.⌋

[constr_3586] `EcucConditionFormula.ecucQueryString` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucConditionFormula.ecucQueryString` shall always be defined.⌋

[constr_3587] `EcucValidationCondition.validationFormula` always required

Imposition time: CP: `IT_EcucDef`

⌈The attribute `EcucValidationCondition.validationFormula` shall always be defined.⌋

[constr_3588] `EcucValueCollection.ecuExtract` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucValueCollection.ecuExtract` shall always be defined.⌋

[constr_3589] `EcucModuleConfigurationValues.ecucDefEdition` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucModuleConfigurationValues.ecucDefEdition` shall always be defined.⌋

[constr_3591] `EcucModuleConfigurationValues.definition` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucModuleConfigurationValues.definition` shall always be defined.⌋

[constr_3592] `EcucContainerValue.definition` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucContainerValue.definition` shall always be defined.⌋

[constr_3593] `EcucParameterValue.definition` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucParameterValue.definition` shall always be defined.⌋

[constr_3594] `EcucNumericalParamValue.value` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucNumericalParamValue.value` shall always be defined.⌋

[constr_3595] `EcucTextualParamValue.value` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucTextualParamValue.value` shall always be defined.⌋

[constr_3596] `EcucAddInfoParamValue.value` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucAddInfoParamValue.value` shall always be defined.⌋

[constr_3597] `EcucAbstractReferenceValue.definition` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucAbstractReferenceValue.definition` shall always be defined.⌋

[constr_3598] `EcucInstanceReferenceValue.value` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucInstanceReferenceValue.value` shall always be defined.⌋

[constr_3599] `EcucReferenceValue.value` always required

Imposition time: CP: `IT_EcucVal`

⌈The attribute `EcucReferenceValue.value` shall always be defined.⌋

[constr_3793] Usage of `KeepLocalPduBuffer`

Status: DRAFT
Imposition time: CP: `IT_EcucVal`

⌈All `Pdu`s that belong to the same Pdu flow shall have `KeepLocalPduBuffer` either set to `TRUE` or set to `FALSE`.⌋

[constr_3794] Usage of `PduBufferAlignment`

Status: DRAFT
Imposition time: CP: `IT_EcucVal`

⌈All `Pdu`s that belong to the same Pdu flow shall have `PduBufferAlignment` set to the same value.⌋

[constr_4200] Existence of `EcucDefinitionCollection.module`

Imposition time: CP: `IT_EcucDef`

⌈`EcucDefinitionCollection.module` shall exist at least once.⌋

[constr_4201] Existence of `EcucModuleDef.container`

Imposition time: CP: `IT_EcucDef`

⌈The reference `EcucModuleDef.container` shall exist at least once.⌋

[constr_4202] Existence of `EcucValueCollection.ecucValue`

Imposition time: CP: `IT_EcucVal`

⌈`EcucValueCollection.ecucValue` shall exist at least once.⌋

[constr_4203] `EcucModuleConfigurationValues.container` ⌈`EcucModuleConfigurationValues.container` shall exist at least once⌋

[constr_5059] Ordering of `MetaDataItem`s of a `MetaDataType`

Imposition time: CP: `IT_EcucVal`

⌈The `MetaDataItem`s of a `MetaDataType` shall be ordered according to their `MetaDataItemLength`. `MetaDataItem`s with greater `MetaDataItemLength` going first.⌋

[constr_5108] `CddModuleId` range restriction

Imposition time: CP: `IT_EcucVal`

⌈The range of `CddModuleId` is restricted to the value 255 and to the range of values 2048..4095.⌋

[constr_5325] Existence of `upperMultiplicityInfinite` and `upperMultiplicity` is mutually exclusive

Imposition time: CP: `IT_EcucDef`

⌈The existence of the elements `upperMultiplicityInfinite` and `upperMultiplicity` shall be mutually exclusive.⌋

[constr_5342] `EcucDefinitionElement.upperMultiplicity` or `EcucDefinitionElement.upperMultiplicityInfinite` required

Imposition time: CP: `IT_EcucDef`

⌈Exactly one of the attributes `EcucDefinitionElement.upperMultiplicity` or `EcucDefinitionElement.upperMultiplicityInfinite` shall be defined.⌋

[constr_5345] Restriction for a reference destination in case of multiple aggregated `EcucParamConfContainerDef`s

Imposition time: CP: `IT_EcucDef`

⌈An `EcucReferenceDef` or `EcucChoiceReferenceDef` is not allowed to reference an `EcucParamConfContainerDef` as destination if

- this `EcucParamConfContainerDef` is aggregated by several `EcucParamConfContainerDef`s as `subContainer` and
- the `EcucParamConfContainerDef` structures in which the referenced `EcucParamConfContainerDef` is aggregated are different compared to the `EcucParamConfContainerDef` structure in which the `EcucReferenceDef` or `EcucChoiceReferenceDef` is located in.

⌋

[constr_5365] Origin information in parameter and reference definitions

Imposition time: CP: `IT_EcucDef`

⌈Each instance of the subclass of `EcucCommonAttributes` or `EcucContainerDef` shall provide a value for the `origin` attribute and this attribute shall be either:

- `'AUTOSAR_ECUC'` - in case that the parameter definition is standardized by AUTOSAR
- vendor specific value - in case that the parameter definition is vendor specific. For vendor specific origins no rules are defined by AUTOSAR and the vendor is free to choose the value (e.g. `'VendorXYZ_v1.3'`).

⌋

[constr_5505] Configuration class of the elements of the `EcucQueryExpression`

Imposition time: CP: `IT_EcucDef`

⌈The elements of the `EcucQueryExpression` involved in one calculation formula shall have lower or equal configuration class (where `PreCompile` configuration class is considered to be the lowest and `PostBuild` the highest) with respect to the context element in which the calculation is performed (e.g. a `Link` configuration parameter can not calculate its value based on a `PostBuild` parameters value).⌋

[constr_5506] Existence of `postBuildVariantMultiplicity` attribute ⌈For each `EcucContainerDef` for which the `upperMultiplicity` is greater than the `lowerMultiplicity`, the attribute `postBuildVariantMultiplicity` shall exist.⌋

[constr_5507] Value of `EcucContainerDef.postBuildVariantMultiplicity` if `postBuildVariantSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildVariantSupport` is set to `false`, every `EcucContainerDef` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildVariantMultiplicity` attribute set to `false`.⌋

[constr_5509] Value of `postBuildVariantMultiplicity` if `postBuildVariantSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildVariantSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildVariantMultiplicity` attribute set to `false`.⌋

[constr_5510] Value of `postBuildVariantValue` if `postBuildVariantSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildVariantSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` shall have its `postBuildVariantValue` attribute set to `false`.⌋

[constr_5520] Latest configuration time to change the values of `symbolicNameValue` parameters

Imposition time: CP: `IT_EcucDef`

⌈The latest configuration time to change the values of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall be `PreCompile` with the setting of `postBuildLoadableValue` and `postBuildVariantValue` to false.⌋

[constr_5521] Latest configuration time to change the number of instances of `symbolicNameValue` parameters ⌈The latest configuration time to change the number of instances of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall be `PreCompile` with the setting of `postBuildLoadableMultiplicity` and `postBuildVariantMultiplicity` to false.⌋

[constr_9439] Existence of attribute `EcucModuleDef.postBuildVariantSupport`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucModuleDef`, the attribute `postBuildVariantSupport` shall exist.⌋

[constr_9440] Existence of attribute `EcucModuleDef.postBuildLoadableSupport`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucModuleDef`, the attribute `postBuildLoadableSupport` shall exist.⌋

[constr_9441] Value of `EcucContainerDef.postBuildLoadableMultiplicity` if `postBuildLoadableSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildLoadableSupport` is set to `false`, every `EcucContainerDef` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildLoadableMultiplicity` attribute set to `false`.⌋

[constr_9442] Value of `postBuildLoadableMultiplicity` if `postBuildLoadableSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildLoadableSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildLoadableMultiplicity` attribute set to `false`.⌋

[constr_9443] Value of `postBuildLoadableValue` if `postBuildLoadableSupport` is set to `false`

Imposition time: CP: `IT_EcucDef`

⌈If `postBuildLoadableSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` shall have its `postBuildLoadableValue` attribute set to `false`.⌋

[constr_9444] Mutual exclusivity of `linkTimeSupport` and `postBuildLoadableMultiplicity`

Imposition time: CP: `IT_EcucDef`

⌈The attributes `EcucContainerDef.linkTimeSupport` and `EcucContainerDef.postBuildLoadableMultiplicity` are mutually exclusive, meaning zero or just one of them shall be used.⌋

[constr_9445] `EcucContainerDef.linkTimeSupport` not applicable in StMD

Imposition time: CP: `IT_EcucDef`

⌈The `EcucContainerDef.linkTimeSupport` attribute shall not be used by an `EcucContainerDef` that is aggregated by an `EcucModuleDef` with the `category` *STANDARDIZED_MODULE_DEFINITION*.⌋

[constr_9446] Existence of the postBuildLoadableMultiplicity attribute

Imposition time: CP: IT_EcucDef

⌈For each EcucContainerDef for which the upperMultiplicity is greater than the lowerMultiplicity, the attribute postBuildLoadableMultiplicity shall exist.⌋

[constr_9447] Removing an instance of the EcucContainerDef at post-build time

Imposition time: CP: IT_EcucDef

⌈Only instances of EcucContainerDefs with postBuildLoadable-Multiplicity or postBuildVariantMultiplicity set to true which are not referenced or are exclusively referenced by EcucAbstractReferenceDefs with postBuildLoadableValue or postBuildVariantValue set to true and have been introduced at post-build time (not part of the initial configuration before post-build updates) can be removed at post-build time.⌋

[constr_9450] EcucCommonAttributes.linkTimeSupport not applicable in St MD

Imposition time: CP: IT_EcucDef

⌈The EcucCommonAttributes.linkTimeSupport attributes shall not be used in EcucCommonAttributes that is aggregated by an EcucModuleDef with the category *STANDARDIZED_MODULE_DEFINITION*.⌋

[constr_9451] Existence of attribute EcucCommonAttributes.linkTime-Support

Imposition time: CP: IT_EcucDef

⌈For each EcucModuleDef with the category *VENDOR_SPECIFIC_MODULE_DEFINITION*, the attribute linkTimeSupport shall exist.⌋

[constr_9452] Mutual exclusivity of linkTimeSupport and postBuild-LoadableMultiplicity

Imposition time: CP: IT_EcucDef

⌈The attributes EcucCommonAttributes.linkTimeSupport and EcucCommon-Attributes.postBuildLoadableMultiplicity are mutually exclusive, meaning zero or just one of them shall be used.⌋

[constr_9453] Mutual exclusivity of linkTimeSupport and postBuild-LoadableValue

Imposition time: CP: IT_EcucDef

⌈The attributes EcucCommonAttributes.linkTimeSupport and EcucCommon-Attributes.postBuildLoadableValue are mutually exclusive, meaning zero or just one of them shall be used.⌋

[constr_9454] Existence of attribute `EcucCommonAttributes.postBuildVariantValue`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucCommonAttributes`, the attribute `postBuildVariantValue` shall exist.⌋

[constr_9455] Existence of attribute `EcucCommonAttributes.postBuildVariantMultiplicity`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucCommonAttributes` for which the `upperMultiplicity` is greater than the `lowerMultiplicity`, the attribute `postBuildVariantMultiplicity` shall exist.⌋

[constr_9456] Existence of attribute `EcucCommonAttributes.postBuildLoadableValue`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucCommonAttributes`, the attribute `postBuildLoadableValue` shall exist.⌋

[constr_9457] Existence of attribute `EcucCommonAttributes.postBuildLoadableMultiplicity`

Imposition time: CP: `IT_EcucDef`

⌈For each `EcucCommonAttributes` for which the `upperMultiplicity` is greater than the `lowerMultiplicity`, the attribute `postBuildLoadableMultiplicity` shall exist.⌋

[constr_9458] `EcucModuleDef` that relies on `EcucCommonAttributes` with `postBuildLoadableValue` set to true of another `EcucModuleDef`

Imposition time: CP: `IT_EcucDef`

⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `postBuildLoadableValue` set to true of another `EcucModuleDef`, the values of these `EcucCommonAttributes` can only differ in different post-build loadable configurations if the implementation of the using `EcucModuleDef` has the `EcucModuleDef.postBuildLoadableSupport` set to `true`.⌋

[constr_9459] `EcucModuleDef` that relies on `EcucCommonAttributes` with `postBuildLoadableMultiplicity` set to true of another `EcucModuleDef`

Imposition time: CP: `IT_EcucDef`

⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `postBuildLoadableMultiplicity` set to true of another `EcucModuleDef`, the number of instances of these `EcucCommonAttributes` can only differ in different post-build loadable configurations if the implementation of the using `EcucModuleDef` has the `EcucModuleDef.postBuildLoadableSupport` set to `true`.⌋

[constr_13000] Existence of reference in the role `EcucContainerDef.destinationUri` for an `EcucContainerDef` aggregated in the role `EcucDestinationUriPolicy.container`

Imposition time: CP: `IT_EcucDef`

⌈If an `EcucContainerDef` is **exclusively** aggregated in the role `EcucDestinationUriPolicy.container`, then it shall not define a reference in the role `destinationUri`.⌋

2.4 CP_TPS_ECUResourceTemplate

[constr_3500] `category` of `HwAttributeDef` shall not be extended ⌈In contrast to the general rule that `category` can be extended by user-specific values it is **not allowed** to extend the meaning of the attribute `category` of meta-class `HwAttributeDef`⌋

[constr_3511] `HwType` shall not have a reference to another `HwType` ⌈A `HwType` (being a `HwDescriptionEntity`) shall not have a reference to another `HwType` in the role `hwType`. The definition of `HwType`s is not hierarchical.⌋

[constr_3512] No support of multiple instantiation ⌈An essential constraint is that each `HwElement` can only be target of one `nestedElement` reference. This means that there is no concept of multiple instantiation of hardware elements. If the same hardware element shall be used several times (using the `nestedElement` reference) each occurrence has to have its own description. This is also true for nested elements of the referenced nested element.⌋

[constr_3513] Scope of connections ⌈Each hardware connection shall only connect features which both are in the hierarchical scope of the hardware element. The hierarchical scope encloses

- all features belonging to the hardware element containing the connection
- all features belonging to hardware elements which are referenced directly and indirectly in the `nestedElement` relation from the hardware element containing connection.

⌋

[constr_11001] Multiplicity of `HwAttributeValue.hwAttributeDef` ⌈For each `HwAttributeValue` the reference in the role `hwAttributeDef` shall exist.⌋

[constr_11002] Multiplicity of `HwElementConnector.hwElement` ⌈For each `HwElementConnector` there shall exist exactly 2 references in the role `hwElement`.⌋

[constr_11003] Multiplicity of `HwPinGroupConnector.hwPinGroup` ⌈For each `HwPinGroupConnector` there shall exist exactly 2 references in the role `hwPinGroup`.⌋

[constr_11004] Multiplicity of `HwPinConnector.hwPin` ⌈For each `HwPinConnector` there shall exist exactly 2 references in the role `hwPin`.⌋

[constr_11005] Multiplicity of `HwAttributeDef.isRequired` ⌈For each `HwAttributeDef` the attribute `isRequired` shall exist.⌋

2.5 CP_TPS_SoftwareComponentTemplate

[constr_1004] Mapping of `ApplicationDataType`s in the scope of single `AtomicSwComponentType`s and `ParameterSwComponentType`s

Imposition time: CP: `IT_CpgExe`

⌈In the scope of `AtomicSwComponentType.internalBehavior.dataTypeMapping` or `ParameterSwComponentType.dataTypeMapping`, each `ApplicationDataType` shall be mapped to exactly one `ImplementationDataType`.⌋

[constr_1005] Compatibility of `ImplementationDataType`s mapped to the same `ApplicationDataType`

Imposition time: CP: `IT_CpgExe`

⌈It is required that `ImplementationDataType`s which are taken for connecting corresponding elements of `PortInterface`s and thus refer to compatible `ApplicationDataType`s are also compatible with each other (so that the RTE is able to cope with possible connections by converting the data accordingly).⌋

[constr_1006] Applicable data categories, depending on specific model elements related to data definition properties

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈

Category	Applicable to ...											Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
VALUE			x	x	x	x		x	x	x	x	x	x	x	x	Contains a single value.
VAL_BLK			x	x	x	x	x				x	x		x		A value block defines values stored together within one calibration parameter object. It is similar to an value array but it stores the values by means of an axis instead (only important for calibration data handling).
DATA_REFERENCE								x	x	x				x[4]	x	Contains an address of another `DataPrototype` (whose type is given via `SwDataDefProps.swPointerTargetProps`).
FUNCTION_REFERENCE								x	x	x					x	Contains an address of a function prototype (whose signature is given via `SwDataDefProps.swPointerTargetProps.functionPointerSignature`).
TYPE_REFERENCE								x	x	x				x	x	The element is defined via reference to another data type (via `SwDataDefProps.implementationDataType`).
STRUCTURE		x		x	x			x	x		x	x	x	x	x	Holds one or several further elements which can have different `AutosarDataType`s. The underlying elements are defined in the same manner as normal data except for the association to `SwAddrMethod`: This has to be the same for all underlying elements. Corresponds to a Record if used in the application domain.
UNION								x	x		x	x	x	x	x	Can hold values of different data types. It is similar to `STRUCTURE` except that all of its members start at the same location in memory. A `UNION` data prototype can contain only one of its elements at a time. The size of the `UNION` is at least the size of the largest member. Please find more information in [TPS_SWCT_01700].
ARRAY	x			x	x		x	x	x		x	x	x	x	x	An array of sub-elements which are of the same type.
BIT											x	x	x		x	One or several bits within a host variable, which are treated as an own data object.

▽

[4] [constr_1295] applies!

△

Category	Applicable to ...											Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
HOST											x	x	x		x	A HOST data type is like a simple VALUE, but it is used for packed bit definition. That means it can host several BIT variables which have their own description and measurement access.
STRING			x	x	x	x					x	x	x	x		Contains a single value interpreted as a text string (note that it appears as a single value for the application domain; the internal representation can be an array).
BOOLEAN			x	x	x	x					x	x	x	x		Contains one boolean state. Depending on the CPU direct addressing of single bits may not be available. So a byte or a word can be used to store only one logical state.
COM_AXIS			x	x	x	x	x				x	x		x		An axis definition as separate calibration parameter which can be referenced by any CURVE, MAP, CUBOID, CUBE_4, and CUBE_5. The benefits by using a common axis is that it saves memory space; because it is stored only one time and can be used in multiple CURVEs, MAPs, CUBOIDs, CUBE_4s, and CUBE_5s.
RES_AXIS			x	x	x	x	x				x	x		x		A RES_AXIS (rescale axis) is also a shared axis like COM_AXIS, the difference is that this kind of axis can be used for rescaling. Note that the RES_AXIS is by nature a CURVE which is used to implement a non linear scaling (rescale) of the axis. In addition to saving memory space via the shared usage like a COM_AXIS, it can compress a huge range to a non-linear distributed axis points thus retaining the required accuracy.
CURVE			x	x	x	x	x				x	x		x		Calibration parameter with one input value and one output value. That means output values can be defined depending on the input value. The granularity of implemented functionality can be changed by using different number of axis points. A CURVE has always one input axis and one output axis. The output axis is a characteristic of the curve and every time present but the input axis can be defined within the curve definition or separately.

▽

△

Category	Applicable to ...											Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
MAP			x	x	x	x	x				x	x		x		Calibration parameter with two input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for y- and x-axis. A MAP has always two input axes and one output axis. The output axis is a characteristic of the MAP and every time present but the input axes can be defined within the MAP definition or separately.
CUBOID			x	x	x	x	x				x	x		x		Calibration parameter with three input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBOID has always three input axes and one output axis. The output axis is a characteristic of the CUBOID and every time present but the input axes can be defined within the CUBOID definition or separately.
CUBE_4			x	x	x	x	x				x	x		x		Calibration parameter with four input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBE_4 has always four input axes and one output axis. The output axis is a characteristic of the CUBE_4 and every time present but the input axes can be defined within the CUBE_4 definition or separately.
CUBE_5			x	x	x	x	x				x	x		x		Calibration parameter with five input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBE_5 has always five input axes and one output axis. The output axis is a characteristic of the CUBE_5 and every time present but the input axes can be defined within the CUBE_5 definition or separately.

⌋

[constr_1007] Allowed attributes of SwDataDefProps for ApplicationDataTypes

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

Attributes of SwDataDefProps	Attribute Existence per ApplicationDataType.category												
	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
additionalNativeTypeQualifier													
annotation	*	*	*	*	*	*	*	*	*	*	*	*	*
baseType													
compuMethod	0..1	0..1				0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.physConstrs	0..1	0..1		0..1		0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.internalConstrs	d/c[5]	d/c		d/c		d/c			d/c	d/c	d/c	d/c	d/c
displayFormat	0..1	0..1		0..1	0..1	0..1			0..1	0..1	0..1	0..1	0..1
displayPresentation	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
implementationDataType													
invalidValue	0..1				0..1								
stepSize	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAddrMethod	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment													
swBitRepresentation													
swCalibrationAccess	0..1	0..1	0..1	0..1	0..1	0..1	1	1	1	1	1	1	1
swCalprmAxisSet							1	1	1	1	1	1	1
swComparisonVariable													
swDataDependency													
swHostVariable													
swImplPolicy	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution	0..1												
swInterpolationMethod	0..1						0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIsVirtual													
swPointerTargetProps													
swRecordLayout	0..1	0..1[6]			0..1		1	1	1	1	1	1	1
swRefreshTiming	0..1	0..1			0..1	0..1							
swTextProps					1								
swValueBlockSize		1											
swValueBlockSizeMult		1											
unit	0..1	0..1							0..1	0..1	0..1	0..1	0..1
valueAxisDataType		0..1					0..1	0..1	0..1	0..1	0..1	0..1	0..1

⌋

[5] don't care
[6] This is required by [TPS_SWCT_01179].

[constr_1009] SwDataDefProps applicable to ImplementationDataTypes

Imposition time: CP: IT_CpgExe

⌈

Attributes of SwDataDefProps	Root Element				Attribute Existence per Category						
	ImplementationDataType	ImplementationDataTypeElement	SwPointerTargetProps	SwServiceArg	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
additionalNativeTypeQualifier	x	x	x	x	0..1	0..1	0..1	0..1	0..1	0..1	0..1
annotation	x	x	x	x	*	*	*	*	*	*	*
baseType	x	x	x	x	1						
compuMethod	x	x	x	x	0..1			0..1			
dataConstr.dataConstrRule.physConstrs	x	x	x	x	d/c[7]			d/c			d/c
dataConstr.dataConstrRule.internalConstrs	x	x	x	x	0..1			0..1			0..1
displayFormat	x	x			0..1				0..1	0..1	0..1
displayPresentation	x	x			0..1						0..1
implementationDataType	x	x	x	x				1			
invalidValue	x	x	x		0..1			0..1	0..1[8]		0..1[9]
stepSize	x	x			0..1						
swAddrMethod	x	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment	x				0..1	0..1	0..1		0..1	0..1	0..1
swBitRepresentation											
swCalibrationAccess	x	x			0..1			0..1	0..1	0..1	0..1
swCalprmAxisSet											
swComparisonVariable											
swDataDependency											
swHostVariable											
swImplPolicy	x		x	x	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution											
swInterpolationMethod											
swIsVirtual											
swPointerTargetProps	x	x	x	x		1	1				

▽

[7] don't care

[8] There is a use case for the definition of an invalidValue for category ARRAY and therefore category STRUCTURE is also supported for the sake of symmetry.

[9] This represents an exception such that it would make sense to use an entire ArrayValue-Specification as the invalidValue because a string semantically is more than just a bunch of characters in a row.

△

Attributes of SwDataDefProps	Root Element				Attribute Existence per Category						
	ImplementationDataType	ImplementationDataTypeElement	SwPointerTargetProps	SwServiceArg	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
swPointerTargetProps .swDataDefProps	x	x	x	x		1					
swPointerTargetProps .functionPointerSignature	x	x	x	x			1				
swRecordLayout											
swRefreshTiming	x	x	x	x	0..1				0..1	0..1	0..1
swTextProps											
swValueBlockSize											
swValueBlockSizeMult											
unit											
valueAxisDataType											

⌋

[constr_1010] If nativeDeclaration does not exist

Imposition time: CP: IT_CpgExe

⌈If nativeDeclaration does not exist in the SwBaseType, it is required that the shortName (e.g. “uint8”) of the corresponding ImplementationDataType is equal to one of the names of the Platform or Standard Types predefined in AUTOSAR code.⌋

[constr_1011] category of SwBaseType

Imposition time: CP: IT_CpgExe

⌈For the attribute SwBaseType.category only the values FIXED_LENGTH and VOID are supported.⌋

[constr_1012] Value of category is FIXED_LENGTH

Imposition time: CP: IT_CpgExe

⌈If

- the value of the attribute SwBaseType.category is set to FIXED_LENGTH and
- the SwBaseType is **not** referenced in the role ApplicationPrimitiveDataType.swDataDefProps.swTextProps.baseType,

then the attribute baseTypeSize shall be filled with content.⌋

[constr_1015] Prioritization of SwDataDefProps

Imposition time: CP: IT_CpgExe

⌈

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory	ComSpec networkRepresentation
additionalNativeTypeQualifier	x		x		D	I			D		S			
annotation			x	D	A	A	A	A	D		A	D		A
baseType	x	x	x		D	I	I	I	D		S	M		D
compuMethod	x	x	x	D	AI	I	I		I	AI	S	D		D
dataConstr	x	x	x	D	C	D	D	I	D		S	D		
displayFormat		x		D	D	D	D	I	D		S	D		
displayPresentation	x	x	x	D	D	D	D				S			
implementationDataType	x		x		D	I	I	I	D					
invalidValue	x	x		D	D	I	I				S			D
stepSize		x		D	D	D	D	D		D	S			
swAddrMethod	x	x	x	D	D	D	D			D			D	
swAlignment	x		x		D	D	D							
swBitRepresentation		x	x								D			
swCalibrationAccess	x	x		D	D	D	D		D	D	S	D		
swCalprmAxisSet	x	x		D		I	I	I			S			
swCalprmAxisSet.swCalprmAxis /SwAxisGrouped.swCalprmRef		x					D	D			S			
swCalprmAxisSet.swCalprmAxis /SwAxisIndividual.swVariableRef		x					D	D			S			
swCalprmAxisSet.swCalprmAxis /SwAxisGrouped.sharedAxisType		x		D							S			
swCalprmAxisSet.swCalprmAxis /SwAxisIndividual.inputVariableType		x		D							S			
swCalprmAxisSet/SwAxisIndividual.unit		opt.		D		I	I	I	I		S			
swComparisonVariable		x						D			S			
swDataDependency		x	x			D	D				S			
swHostVariable		x	x								D			
swImplPolicy	x		x	D	D	D			D					

▽

△

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory	ComSpec networkRepresentation
swIntendedResolution			x	D[10]										
swInterpolationMethod			x	D	I	D	D	D			S			
swIsVirtual		x				D	D				S			
swPointerTargetProps			x		D	I			D					
swRecordLayout	x	x	x	D		I	I	I			S			
swRefreshTiming		x		D	D	D	D		D	D	D			
swTextProps		x	x	D	I	I	I	I			S			
swValueBlockSize		x	x	D	I	I	I	I			S			
swValueBlockSizeMult		x	x	D	I	I	I	I			S			
unit		x	x	D	I	I	I		I		S	D		
valueAxisDataType		x	x	D	I	I	I	I			S			

⌋

[constr_1016] Restriction of invalidValue for ImplementationDataType and ImplementationDataTypeElement

Imposition time: CP: IT_CpgExe

⌈invalidValue for ImplementationDataType and ImplementationDataTypeElement is restricted to either a compatible NumericalValueSpecification, TextValueSpecification (caution, [constr_1284] applies) or a ConstantReference that in turn points to a compatible ValueSpecification.⌋

[10] swIntendedResolution is used only in an early phase of the definition of data types, namely in the context of the definition of so-called blueprints. To that extent, swIntendedResolution represents a non-binding requirement that shall later be considered for the definition of an appropriate CompuMethod.

[constr_1017] Supported combinations of swImplPolicy and swCalibrationAccess

Imposition time: CP: IT_CpgExe

⌈

swImplPolicy	swCalibrationAccess		
	notAccessible	readOnly	readWrite
fixed	**yes**	not supported	not supported
const	**yes**	**yes**	not supported
standard	**yes**	**yes**	**yes**
queued	**yes**	not supported	not supported
measurementPoint	not supported	**yes**	not supported

⌋

[constr_1018] dataElement with swImplPolicy set to measurementPoint shall not be referenced by a VariableAccess aggregated by RunnableEntity in the role dataReadAccess

Imposition time: CP: IT_CpgExe

⌈Due to the nature of dataElements with swImplPolicy set to measurementPoint, such dataElements shall not be referenced by a VariableAccess aggregated by RunnableEntity in the role dataReadAccess.⌋

[constr_1020] ParameterDataPrototype shall be of compatible data type as referenced in sharedAxisType

Imposition time: CP: IT_CpgExe

⌈The ParameterDataPrototype assigned in swCalprmRef shall be typed by data type compatible with sharedAxisType.⌋

[constr_1022] Limits shall be defined for each direction of CompuMethod

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈In the scope of a CompuMethod, if both aggregations in the role compuInternalToPhys and compuPhysToInternal exist, then for all

- compuInternalToPhys.compuScale and
- compuPhysToInternal.compuScale

the attributes lowerLimit and upperLimit shall exist.⌋

[constr_1024] Stepwise definition of CompuMethods

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈In a bound model, the intervals (i.e. determined by attributes CompuScale.lowerLimit and CompuScale.upperLimit) defined by CompuScales used in the context of a given CompuMethod of all values of category except BITFIELD_TEXTTABLE shall **not** overlap.

For CompuMethods of category BITFIELD_TEXTTABLE, the combination of the interval created by attributes CompuScale.upperLimit, CompuScale.lowerLimit and CompuScale.mask shall be unique in the context of the enclosing CompuMethod.⌋

[constr_1026] Compatibility of Units

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If a SwDataDefProps references a Unit and the SwDataDefProps has a reference to either of/or both

- a CompuMethod that in turn references a Unit
- a DataConstr that in turn references a Unit

then the Units referenced from

- SwDataDefProps
- SwDataDefProps.compuMethod
- SwDataDefProps.dataConstr

shall be compatible.⌋

[constr_1029] ConstantSpecificationMapping and ConstantSpecification

Imposition time: CP: IT_CpgExe

⌈It is required that the ConstantSpecification referenced from a ConstantSpecificationMapping in the role applConstant shall fulfill the criteria defined in [TPS_SWCT_01871] (i.e. be defined in the *application domain*, applConstant) and the other ConstantSpecification referenced in the role implConstant shall fulfill the criteria defined in [TPS_SWCT_01872] (i.e. be defined in the *implementation domain*, implConstant).⌋

[constr_1033] Communication scenarios for sender/receiver communication

Imposition time: CP: IT_CompSwcT

⌈For sender/receiver communication, it is not allowed to create a communication scenario where n senders are connected to m receivers, where m and n are **both** greater than 1.⌋

[constr_1035] Recursive definition of CompositionSwComponentType

Imposition time: CP: IT_CompSwcT, AP: IT_BefAraApiGen

⌈The recursive definition of a CompositionSwComponentType that eventually contains a SwComponentPrototype typed by the same CompositionSwComponentType shall not be feasible.⌋

[constr_1036] Connect kinds of PortInterfaces

Imposition time: CP: IT_RteGen

⌈It shall not be possible to connect PortPrototypes typed by PortInterfaces of different kinds.

Subclasses of DataInterface make an exception to this rule and can be used for creating connections to each other.⌋

[constr_1037] Client shall not be connected to multiple servers

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈A client shall not be connected to multiple servers such that an operation call would be handled by more than one server.⌋

[constr_1038] Reference to ApplicationError

Imposition time: CP: IT_CpgExe

⌈A possibleError referenced by a ClientServerOperation shall be owned by the ClientServerInterface that also owns the ClientServerOperation.⌋

[constr_1039] Relevance of swImplPolicy

Imposition time: CP: IT_RteGen

⌈It is not possible to define a mapping between an element where the swImplPolicy is set to queued and another element where the swImplPolicy is set differently.⌋

[constr_1043] Allowed combinations of a specific *Type of PortInterface*, a specific *Type of PortPrototype*, and a specific *Type of ComSpec*

Imposition time: CP: IT_CpgExe

⌈

Type of PortPrototype	Type of ComSpec	Role of Element	Type of PortInterface	Role of Type-Ref
PPortPrototype	NonqueuedSender-ComSpec	dataElement	SenderReceiver-Interface	providedInterface
PPortPrototype	QueuedSenderCom-Spec	dataElement	SenderReceiver-Interface	providedInterface
RPortPrototype	Nonqueued-ReceiverComSpec	dataElement	SenderReceiver-Interface	requiredInterface
RPortPrototype	QueuedReceiver-ComSpec	dataElement	SenderReceiver-Interface	requiredInterface
PRPortPrototype	NonqueuedSender-ComSpec	dataElement	SenderReceiver-Interface	providedRequired-Interface
PRPortPrototype	Nonqueued-ReceiverComSpec	dataElement	SenderReceiver-Interface	providedRequired-Interface
PRPortPrototype	QueuedReceiver-ComSpec	dataElement	SenderReceiver-Interface	providedRequired-Interface
PRPortPrototype	QueuedSenderCom-Spec	dataElement	SenderReceiver-Interface	providedRequired-Interface
PPortPrototype	NvProvideComSpec	nvData	NvDataInterface	providedInterface

▽

△

Type of PortPrototype	Type of ComSpec	Role of Element	Type of PortInterface	Role of Type-Ref
RPortPrototype	NvRequireComSpec	nvData	NvDataInterface	requiredInterface
PRPortPrototype	NvProvideComSpec	nvData	NvDataInterface	providedRequired-Interface
PRPortPrototype	NvRequireComSpec	nvData	NvDataInterface	providedRequired-Interface
PPortPrototype	ModeSwitchSender-ComSpec	modeGroup	ModeSwitch-Interface	providedInterface
RPortPrototype	ModeSwitch-ReceiverComSpec	modeGroup	ModeSwitch-Interface	requiredInterface
PRPortPrototype	ModeSwitchSender-ComSpec	modeGroup	ModeSwitch-Interface	providedRequired-Interface
PRPortPrototype	ModeSwitch-ReceiverComSpec	modeGroup	ModeSwitch-Interface	providedRequired-Interface
PPortPrototype	ParameterProvide-ComSpec	parameter	Parameter-Interface	providedInterface
RPortPrototype	ParameterRequire-ComSpec	parameter	Parameter-Interface	requiredInterface
PPortPrototype	ServerComSpec	operation	ClientServer-Interface	providedInterface
RPortPrototype	ClientComSpec	operation	ClientServer-Interface	requiredInterface
PRPortPrototype	ServerComSpec	operation	ClientServer-Interface	providedRequired-Interface
PRPortPrototype	ClientComSpec	operation	ClientServer-Interface	providedRequired-Interface

⌋

[constr_1044] Applicability of DataFilter

Imposition time: CP: IT_CpgExe

⌈According to the origin of DataFilter, i.e. [5, ISO 17356-4], DataFilters can only be applied to values with an integer base type.⌋

[constr_1046] Applicability of [TPS_SWCT_01845]

Imposition time: CP: IT_CpgExe

⌈[TPS_SWCT_01845] applies **only** if the value of the attribute isService is set to false.⌋

[constr_1047] Compatibility of ApplicationPrimitiveDataTypes

Imposition time: CP: IT_RteGen

⌈Instances of ApplicationPrimitiveDataType are compatible if and only if one of the following conditions applies:

1. All the following sub conditions apply:
   (a) They have the same category.
   (b) The swDataDefProps (after consideration of [constr_1015]) attached to the M1 data types are compatible.

2. In the context of using the ApplicationPrimitiveDataType, a DataPrototypeMapping exists that refers to a DataPrototype typed by one of the ApplicationPrimitiveDataTypes in the role firstDataPrototype and to another DataPrototype typed by the other ApplicationPrimitiveDataType in the role secondDataPrototype.
3. In the context of using the ApplicationPrimitiveDataType, a DataPrototypeMapping exists that refers to a DataPrototype typed by the ApplicationPrimitiveDataType in the role secondDataPrototype and to another DataPrototype typed by an ApplicationCompositeDataType in the role firstDataPrototype and additionally for the side of the ApplicationCompositeDataType a corresponding ApplicationCompositeDataTypeSubElementRef exists in the role firstElement that in turn references an ApplicationCompositeElementDataPrototype.

⌋

[constr_1048] Compatibility of ApplicationRecordDataTypes

Imposition time: CP: IT_RteGen

⌈Instances of ApplicationRecordDataTypes are compatible if and only if one of the following conditions applies:

1. All elements *at the same record position* are of compatible AutosarDataTypes (either ApplicationCompositeDataTypes or ApplicationPrimitiveDataTypes).
2. For each ApplicationRecordDataType.element, the attribute isOptional shall either
   - not exist on both sides or
   - be set to the value false if it only exists on one side or
   - have the identical value on both sides.
3. In the context of a DataPrototypeMapping, for each ApplicationRecordElement of the required ApplicationRecordDataType a SubElementMapping exists such that a ApplicationCompositeDataTypeSubElementRef in the role firstElement or secondElement exists that references the required ApplicationRecordElement **and** a corresponding ApplicationCompositeDataTypeSubElementRef exists in the **other** role (i.e. secondElement or firstElement) that in turn references an ApplicationRecordElement of the provided ApplicationRecordDataType.

⌋

[constr_1049] Compatibility of ApplicationArrayDataTypes

Imposition time: CP: IT_RteGen

⌈Instances of ApplicationArrayDataType are compatible if and only if one of the following conditions applies:

1. All the following sub conditions apply:
   (a) Their elements are of compatible AutosarDataTypes (either ApplicationCompositeDataTypes or ApplicationPrimitiveDataTypes).
   (b) The attributes maxNumberOfElements and arraySizeSemantics (given the existence) have identical values.
2. In the context of a DataPrototypeMapping, for the ApplicationArrayElement of the required ApplicationArrayDataType a SubElementMapping exists such that a ApplicationCompositeDataTypeSubElementRef in the role firstElement or secondElement exists that references the required ApplicationArrayElement **and** a corresponding ApplicationCompositeDataTypeSubElementRef exists in the **other** role (i.e. secondElement or firstElement) that in turn references an ApplicationArrayElement of the provided ApplicationArrayDataType.

⌋

[constr_1050] Extended compatibility of ImplementationDataTypes

Imposition time: CP: IT_RteGen

⌈Instances of ImplementationDataType are compatible if and only if after all type-references are resolved one of the following rules applies:

1. The two ImplementationDataTypes are compatible according to [constr_10667].
2. In the context of using the ImplementationDataType, a DataPrototypeMapping exists that refers to a DataPrototype typed by one of the ImplementationDataTypes in the role firstDataPrototype and to another DataPrototype typed by the other ImplementationDataType in the role secondDataPrototype.
3. In the context of using the ImplementationDataType, a DataPrototypeMapping exists that refers to a DataPrototype typed by the ImplementationDataType in the role secondDataPrototype and to another DataPrototype typed by an ImplementationDataType with a subElement in the role firstDataPrototype and additionally for the side of the ImplementationDataType with a subElement a corresponding ImplementationDataTypeSubElementRef exists in the role firstElement that in turn references an ImplementationDataTypeElement.

⌋

[constr_1051] Compatibility of SwDataDefProps

Imposition time: CP: IT_RteGen

⌈SwDataDefProps are compatible if and only if:

1. They refer to compatible Unit definitions, or neither of them has an associated Unit.
2. They refer to compatible conversion methods or neither of them associates such a method.
3. They both aggregate a ValueSpecification in the role invalidValue or neither of them aggregates a ValueSpecification in the role invalidValue.
4. If existent (see previous condition), one of the following conditions applies to ValueSpecifications aggregated in the role invalidValue for being considered compatible (after following and resolving indirections created by ConstantReference):
   (a) both are ApplicationValueSpecifications and the values are compatible according to [TPS_GST_02501].
   (b) both are NumericalValueSpecifications and the values are compatible according to [TPS_GST_02501].
   (c) both are TextValueSpecifications and the values are identical.
   (d) both are ArrayValueSpecifications and the values are effectively identical, e.g. if one ArrayValueSpecification specifies all values explicitly and the other ArrayValueSpecification specifies values based on a rule then the yield of both ArrayValueSpecifications (i.e. element for element) shall be identical.
   (e) both are RecordValueSpecifications and the values are identical.
   (f) if one is a NumericalValueSpecification and the other one is an ApplicationValueSpecification then the check for compatibility shall apply the CompuMethod on the physical value such that a comparison on the implementation level becomes possible. [TPS_GST_02501] applies[11].
5. They refer to compatible data constraints dataConstr.
6. They refer to compatible swRecordLayouts
7. They refer to compatible swTextProps, or neither of them references SwTextProps in the role swTextProps.

All other attributes (e.g. swCalibrationAccess do not affect compatibility).⌋

[11] if one is a NumericalValueSpecification and the other one is an ApplicationValueSpecification and the application of the CompuMethod on the side of the ApplicationValueSpecification does not yield a valid number, a comparison is not possible.

[constr_1052] Compatibility of Units

Imposition time: CP: IT_RteGen

⌈Two Unit definitions are compatible if and only if:

1. They have compatible (see [TPS_GST_02501]) values of attributes factorSiToUnit and offsetSiToUnit.
2. One of the following conditions is fulfilled:
   - They refer to compatible definitions of PhysicalDimension.
   - Neither of them associates a PhysicalDimension.
   - One Unit refers to a PhysicalDimension with shortName NoDimension where all exponents are set to 0 and the other Unit does not refer to a PhysicalDimension.

⌋

[constr_1053] Compatibility of PhysicalDimensions in the context of the creation of a SwConnector

Imposition time: CP: IT_RteGen

⌈In the context of the creation of a SwConnector, two PhysicalDimension definitions are compatible if and only if the values of

- lengthExp
- massExp
- timeExp
- currentExp
- temperatureExp
- molarAmountExp
- luminousIntensityExp

are identical and **either**

- the shortNames are identical **or**
- a PhysicalDimensionMapping exists that maps one of the PhysicalDimensions in the role firstPhysicalDimension and the other PhysicalDimension in the role secondPhysicalDimension.

⌋

[constr_1054] No DataConstr available at the provider

Imposition time: CP: IT_RteGen

⌈If the provider defines no constraints, it is only compatible with a receiver which also defines no constraints at all.⌋

[constr_1055] ImplementationDataType has category VALUE

Imposition time: CP: IT_CpgExe

⌈The attributes baseType shall refer to a compatible SwBaseType.⌋

[constr_1056] ImplementationDataType has category TYPE_REFERENCE

Imposition time: CP: IT_CpgExe

⌈The ImplementationDataTypes referenced by the attributes SwDataDefProps.implementationDataType shall be compatible.⌋

[constr_1057] ImplementationDataType has category DATA_REFERENCE

Imposition time: CP: IT_CpgExe

⌈The attributes SwDataDefProps.swPointerTargetProps shall have identical targetCategory and shall refer to SwDataDefProps where all attributes are identical.⌋

[constr_1058] ImplementationDataType has category FUNCTION_REFERENCE

Imposition time: CP: IT_CpgExe

⌈The attributes SwDataDefProps.swPointerTargetProps.functionPointerSignature shall refer to BswModuleEntrys which each resolve to the **same function signature**.⌋

[constr_1059] Compatibility of data types with category VALUE

Imposition time: CP: IT_RteGen

⌈An ApplicationDataType of category VALUE shall (after all indirections created by ImplementationDataTypes of category TYPE_REFERENCE are resolved) only be mapped/connected to an ImplementationDataType which also has category VALUE.⌋

[constr_1060] Compatibility of data types with category ARRAY, VAL_BLK

Imposition time: CP: IT_CpgExe

⌈

ApplicationDataType	ImplementationDataType: Array of uint8	ImplementationDataType: Array of other
ApplicationArrayDataType of category ARRAY, VAL_BLK, arraySizeSemantics = fixedSize	ImplementationDataType of category ARRAY, with ImplementationDataTypeElement with arraySizeSemantics = fixedSize	ImplementationDataType of category ARRAY, with ImplementationDataTypeElement with arraySizeSemantics = fixedSize

▽

△

ApplicationArrayDataType of category ARRAY, VAL_BLK, arraySizeSemantics = variableSize	ImplementationDataType of category ARRAY, with ImplementationDataTypeElement with arraySizeSemantics = variableSize or Variable-Size Array Data Type	Variable-Size Array Data Type

⌋

[constr_1061] Compatibility of data types with category STRUCTURE

Imposition time: CP: IT_CpgExe

⌈An ApplicationDataType of category STRUCTURE shall (after all indirections created by ImplementationDataTypes of category TYPE_REFERENCE are resolved) only be mapped/connected to an ImplementationDataType of category STRUCTURE.⌋

[constr_1063] Compatibility of data types with category BOOLEAN

Imposition time: CP: IT_CpgExe

⌈An ApplicationDataType of category BOOLEAN shall (after all indirections created by ImplementationDataTypes of category TYPE_REFERENCE are resolved) only be mapped/connected to an ImplementationDataType of category VALUE.⌋

[constr_1064] Compatibility of data types with category COM_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, or CUBE_5

Imposition time: CP: IT_CpgExe

⌈An ApplicationDataType of category

- COM_AXIS,
- RES_AXIS,
- CURVE,
- MAP,
- CUBOID,
- CUBE_4, or
- CUBE_5

shall (after all indirections created by ImplementationDataTypes of category TYPE_REFERENCE are resolved) only be mapped/connected to an ImplementationDataType of category

- STRUCTURE or
- ARRAY.

⌋

[constr_1066] Forbidden mappings to `ImplementationDataType`

Imposition time: CP: `IT_CpgExe`

⌈An `ApplicationDataType` shall never be mapped to

- an `ImplementationDataType` of `category`
  - `UNION`,
  - `DATA_REFERENCE`, or
  - `FUNCTION_REFERENCE`,
- or to an `ImplementationDataType` that contains `subElement`s of `category`
  - `UNION`,
  - `DATA_REFERENCE`, or
  - `FUNCTION_REFERENCE`.

⌋

[constr_1068] Compatibility of `VariableDataPrototype`s or `ParameterDataPrototype`s typed by primitive data types

Imposition time: CP: `IT_RteGen`

⌈Two `VariableDataPrototype`s or `ParameterDataPrototype`s of `ApplicationPrimitiveDataType`s or `ImplementationDataType`s of `category` `VALUE`, `BOOLEAN`, or `STRING` are compatible if and only if one of the following conditions applies:

1. All the following subconditions apply:
   (a) They are typed by (read “refer to”) compatible `AutosarDataType`s
   (b) The two `VariableDataPrototype`s or `ParameterDataPrototype`s have identical `shortName`s. This is required to map `VariableDataPrototype`s in unordered `SenderReceiverInterface`s, `NvDataInterface`s and `ParameterInterface`s.
   (c) The attribute `swImplPolicy` is either set to `queued` for both or none of the `VariableDataPrototype`s.
2. In the context of a `DataPrototypeMapping`, one of the applicable `VariableDataPrototype`s or `ParameterDataPrototype`s is referenced by the `DataPrototypeMapping` in the role `firstDataPrototype` and the other `VariableDataPrototype`s or `ParameterDataPrototype`s is referenced by the same `DataPrototypeMapping` in the role `secondDataPrototype`.

⌋

[constr_1069] Compatibility of PortPrototypes of different DataInterfaces in the context of AssemblySwConnectors

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different DataInterfaces are compatible if and only if

1. One of the following conditions applies:
   (a) For each VariableDataPrototype or ParameterDataPrototype defined in the context of the DataInterface of the required PortPrototype a compatible (see [constr_1068]) VariableDataPrototype or ParameterDataPrototype exists in the DataInterface of the provided PortPrototype.

   The shortNames of VariableDataPrototypes and ParameterDataPrototypes are used to identify the pair.

   (b) A VariableAndParameterInterfaceMapping.dataMapping exists for which the following conditions apply:

      i. It is referenced by the corresponding SwConnector.

      ii. It references one of the two VariableDataPrototypes or ParameterDataPrototypes in the role firstDataPrototype and the other in the role secondDataPrototype.

2. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1070] Compatibility of PortPrototypes of different DataInterfaces in the context of DelegationSwConnectors

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different DataInterfaces are compatible if and only if

1. One of the following conditions applies:
   (a) For each VariableDataPrototype or ParameterDataPrototype defined in the context of the DataInterface of the required inner PortPrototype a compatible VariableDataPrototype or ParameterDataPrototype exists in the DataInterface of the required outer PortPrototype.

   The shortName of VariableDataPrototypes and ParameterDataPrototypes are used to identify the pair.

   [constr_1071] defines which PortInterface elements are compatible depending on the PortInterface type and the swImplPolicy attributes of the PortInterface elements.

   (b) A VariableAndParameterInterfaceMapping.dataMapping exists for which the following conditions apply:

i. It is referenced by the corresponding SwConnector.

ii. It references one of the two VariableDataPrototypes or ParameterDataPrototypes in the role firstDataPrototype and the other in the role secondDataPrototype.

2. One of the following conditions applies:

   (a) For at least one VariableDataPrototype or ParameterDataPrototype defined in the context of the SenderReceiverInterface, NvDataInterface or ParameterInterface of the provided inner PortPrototype, a compatible VariableDataPrototype or ParameterDataPrototype exists in the SenderReceiverInterface, NvDataInterface or ParameterInterface of the provided outer PortPrototype.

   The shortNames of VariableDataPrototypes and ParameterDataPrototypes are used to identify the pair.

   [constr_1071] defines which PortInterface elements are compatible depending on the PortInterface type and the swImplPolicy attributes of the PortInterface elements.

   (b) A VariableAndParameterInterfaceMapping.dataMapping exists for which the following conditions apply:

      i. It is (if a corresponding SwConnector already exists) referenced by the corresponding SwConnector.

      ii. It references one of the two VariableDataPrototypes or ParameterDataPrototypes in the role firstDataPrototype and the other in the role secondDataPrototype.

3. For each such pair, the values of their isService attributes are identical.

4. For each such pair, either

   (a) no meta-data are defined on both sides or

   (b) both sides define a SenderReceiverInterface.metaDataItemSet and the content of the aggregated MetaDataItemSet is identical on both sides.

   In this context, “identical” means that the respective MetaDataItemSets define ordered collections of MetaDataItems where the corresponding MetaDataItem.metaDataItemType.value have identical content.

⌋

[constr_1071] compatibility of ParameterDataPrototype and VariableDataPrototype

Imposition time: CP: IT_RteGen

⌈

Provided Port Required Outer Port Provided Inner Port Required Outer Port			Required Port / Required Inner Port / Provided Outer Port / Provided Outer Port					
PortInterface			Prm			S/R		NvD
Interface Element			PDP			VDP		VDP
	SwImplPolicyEnum		fixed	const	standard	standard	queued	standard
Prm	PDP	fixed	yes	yes	yes	yes	no	yes
		const	no	yes	yes	yes	no	yes
		standard	no	no	yes	yes	no	yes
S/R	VDP	standard	no	no	no	yes	no	yes
		queued	no	no	no	no	yes	no
NvD	VDP	standard	no	no	no	yes	no	yes

⌋

[constr_1072] Compatibility of ModeSwitchInterfaces in the context of an AssemblySwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different ModeSwitchInterfaces are compatible if and only if

1. One of the following conditions applies:
   (a) For the ModeDeclarationGroupPrototype defined in the context of the ModeSwitchInterface of the required PortPrototype a compatible ModeDeclarationGroupPrototype exists in the ModeSwitchInterface of the provided PortPrototype.
   (b) A ModeInterfaceMapping.modeMapping exists for which the following conditions apply:
      i. It is referenced by the corresponding SwConnector.
      ii. It references one of the two ModeDeclarationGroupPrototypes in the role firstModeGroup and the other in the role secondModeGroup.
2. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1073] Compatibility of ModeSwitchInterfaces in the context of an DelegationSwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different ModeSwitchInterfaces are compatible if and only if

1. One of the following conditions applies:
   (a) For the ModeDeclarationGroupPrototype defined in the context of the ModeSwitchInterface of the inner PortPrototype a compatible ModeDeclarationGroupPrototype exists in the ModeSwitchInterface of the outer PortPrototype.
   (b) A ModeInterfaceMapping.modeMapping exists for which the following conditions apply:
      i. It is referenced by the corresponding SwConnector.
      ii. It references one of the two ModeDeclarationGroupPrototypes in the role firstModeGroup and the other in the role secondModeGroup.
2. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1074] Compatibility of ModeDeclarationGroupPrototypes

Imposition time: CP: IT_RteGen

⌈ModeDeclarationGroupPrototypes are compatible if and only if one of the following conditions applies:

1. They are typed by (read “refer to”) compatible ModeDeclarationGroups.
2. A ModeDeclarationGroupPrototypeMapping exists that identifies the differently named ModeDeclarationGroupPrototypes that correlate with each other. [constr_1210] applies.

⌋

[constr_1075] Compatibility of ModeDeclarationGroups

Imposition time: CP: IT_RteGen

⌈ModeDeclarationGroups are compatible if and only if one of the following conditions applies:

1. All the following subconditions apply:
   (a) They define an identical number of ModeDeclarations.
   (b) Each ModeDeclaration on the required side corresponds to a ModeDeclaration on the provided side with an identical shortName.

(c) The initialModes on both sides refer to ModeDeclarations with identical shortNames.

(d) The attribute in the role ModeDeclarationGroup.modeUserErrorBehavior.errorReactionPolicy has identical values on both sides.

(e) The attribute in the role ModeDeclarationGroup.modeManagerErrorBehavior.errorReactionPolicy has identical values on both sides.

(f) The attribute in the role ModeDeclarationGroup.modeUserErrorBehavior.defaultMode either does not exist on both sides or refers on both sides to ModeDeclarations with identical shortNames.

(g) The attribute in the role ModeDeclarationGroup.modeManagerErrorBehavior.defaultMode either does not exist on both sides or refers on both sides to ModeDeclarations with identical shortNames.

(h) one of the following subconditions applies:

- the attribute category has the value ALPHABETIC_ORDER on both sides.
- the attribute category has the value EXPLICIT_ORDER on both sides **and** the matching ModeDeclarations according to 1(b) have the identical values of the attributes ModeDeclaration.value **and** also the value of ModeDeclarationGroup.onTransitionValue matches on both sides.

2. A ModeDeclarationMapping is applied which identifies the corresponding ModeDeclarations.

In addition, the compatibility of corresponding ModeTransitions shall be checked, i.e. [constr_1194] and [constr_1245] apply.⌋

[constr_1076] Compatibility of ArgumentDataPrototypes

Imposition time: CP: IT_RteGen

⌈Two ArgumentDataPrototypes are compatible if and only if

1. They are typed by compatible AutosarDataTypes **or** a ClientServerOperationMapping.argumentMapping exists that references one ArgumentDataPrototype in the role firstDataPrototype and the other ArgumentDataPrototype in the role secondDataPrototype.
2. They have the same value of the argument direction (in, out or inout), i.e. [constr_1268] applies.

⌋

[constr_1077] Compatibility of `ApplicationError`s

Imposition time: CP: `IT_RteGen`

⌈Two `ApplicationError`s are compatible if and only if one of the following conditions applies:

1. All the following subconditions apply:
   (a) They have the same `shortName`.
   (b) They have the same attributes. Especially the `errorCode` shall be identical in both `ApplicationError`s.
2. A `ClientServerInterfaceMapping.errorMapping` exists that references one of the `ApplicationError`s in the role `firstApplicationError` and the other `ApplicationError`s in the role `secondApplicationError`.

⌋

[constr_1078] Compatibility of `ClientServerOperation`s

Imposition time: CP: `IT_RteGen`

⌈Two `ClientServerOperation`s are considered compatible if their signatures match. In particular, they are compatible if and only if

1. They have the same number of `ArgumentDataPrototype`s.
2. The n-th arguments of both `ClientServerOperation`s are compatible. This implies an ordering of `ArgumentDataPrototype`s.
3. They have identical values of attribute `diagArgIntegrity` or the attribute `diagArgIntegrity` does not exist on both sides.
4. They have the same `shortName` (again allows for mapping in `PortInterface`s).
5. The required `ClientServerOperation` specifies a compatible `ApplicationError` for each `ApplicationError` that is possibly raised by the provided `ClientServerOperation`, and possibly others. Thereby, `ClientServerOperation`s that refer to a `possibleError` that represents the value `E_OK` are compatible with `ClientServerOperation`s that do refer to `possibleError`s where none of them represents the value `E_OK`.

⌋

[constr_1079] Compatibility of `ClientServerInterface`s in the context of an `AssemblySwConnector`

Imposition time: CP: `IT_RteGen`

⌈`ClientServerInterface`s are compatible if and only if

1. One of the following conditions applies:

(a) For each ClientServerOperation defined in the context of the ClientServerInterface of the required PortPrototype a compatible ClientServerOperation exists in the ClientServerInterface of the provided PortPrototype. The shortNames of ClientServerOperations are used to identify the pair.

(b) A ClientServerInterfaceMapping.operationMapping exists for which the following conditions apply:

i. It is referenced by the corresponding SwConnector.

ii. It references one of the two ClientServerOperations in the role firstOperation and the other in the role secondOperation.

2. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1080] Compatibility of ClientServerInterfaces in the context of an DelegationSwConnector

Imposition time: CP: IT_RteGen

⌈ClientServerInterfaces are compatible if and only if

1. One of the following conditions applies:

(a) For each ClientServerOperation defined in the context of the ClientServerInterface of the required inner PortPrototype a compatible ClientServerOperation exists in the ClientServerInterface of the required outer PortPrototype. The shortNames of ClientServerOperations are used to identify the pair.

(b) A ClientServerInterfaceMapping.operationMapping exists for which the following conditions apply:

i. It is referenced by the corresponding SwConnector.

ii. It references one of the two ClientServerOperations in the role firstOperation and the other in the role secondOperation.

2. One of the following conditions applies:

(a) For at least one ClientServerOperation defined in the context of the ClientServerInterface of the provided inner PortPrototype a compatible ClientServerOperation exists in the ClientServerInterface of the provided outer PortPrototype. The shortNames of ClientServerOperations are used to identify the pair.

(b) A ClientServerInterfaceMapping.operationMapping exists for which the following conditions apply:

i. It is referenced by the corresponding SwConnector.

ii. It references one of the two ClientServerOperations in the role firstOperation and the other in the role secondOperation.

3. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1081] Compatibility of TriggerInterfaces in the context of an AssemblySwConnector

Imposition time: CP: IT_RteGen

⌈TriggerInterfaces are compatible if and only if

1. One of the following conditions applies:
   (a) For each Trigger defined in the context of the TriggerInterface of the required PortPrototype a compatible Trigger exists in the TriggerInterface of the provided PortPrototype. The shortNames of Trigger are used to identify the pair.
   (b) A TriggerInterfaceMapping.triggerMapping exists for which the following conditions apply:
      i. It is referenced by the corresponding SwConnector.
      ii. It references one of the two Triggers in the role firstTrigger and the other in the role secondTrigger.
2. For each such pair, the values of their isService attributes are identical.

⌋

[constr_1082] Compatibility of TriggerInterfaces in the context of an DelegationSwConnector

Imposition time: CP: IT_RteGen

⌈TriggerInterfaces are compatible if and only if all the following conditions apply:

1. One of the following subconditions applies:
   (a) For each Trigger defined in the context of the TriggerInterface of the **required** inner PortPrototype a compatible Trigger exists in the TriggerInterface of the **required** outer PortPrototype. The shortNames of Trigger are used to identify the pair.
   (b) For at least one Trigger defined in the context of the TriggerInterface of the **provided** outer PortPrototype a compatible Trigger exists in the TriggerInterface of the **provided** inner PortPrototype. The shortNames of Trigger are used to identify the pair.
   (c) A TriggerInterfaceMapping.triggerMapping exists for which all the following conditions apply:
      i. It is referenced by the corresponding SwConnector.

ii. It references one of the two `Trigger`s in the role `firstTrigger` and the other in the role `secondTrigger`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋

[constr_1083] Compatibility of `Trigger`s

Imposition time: CP: `IT_RteGen`

⌈`Trigger`s are compatible if one of the following conditions is fulfilled:

- They have an identical `shortName`.
- A `TriggerMapping` exists that references one of the `Trigger`s in the role `firstTrigger` and the other `Trigger` in the role `secondTrigger`.

⌋

[constr_1084] delegation of a provided outer `PortPrototype`

Imposition time: CP: `IT_RteGen`

⌈The delegation of a provided outer `PortPrototype` is properly defined if the following criteria are fulfilled:

1. For each `VariableDataPrototype` or `ParameterDataPrototype` present in the
   - `SenderReceiverInterface`,
   - `NvDataInterface`, or
   - `ParameterInterface`

   of the provided outer `PortPrototype`, at least one connection via
   - `DelegationSwConnector` to a provided inner `PortPrototype` **or**
   - `PassThroughSwConnector` to a required outer `PortPrototype`

   with a compatible `VariableDataPrototype` or `ParameterDataPrototype` in the
   - `SenderReceiverInterface`,
   - `NvDataInterface`, or
   - `ParameterInterface`

   of the
   - provided inner `PortPrototype` **or**
   - required outer `PortPrototype`

   exists.

Either the shortNames of VariableDataPrototypes or ParameterDataPrototypes are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.

2. For each VariableDataPrototype provided by a PRPortPrototype that is typed by a

   - SenderReceiverInterface or
   - NvDataInterface

   and that is referenced in the role outerPort by a DelegationSwConnector, a corresponding VariableDataPrototype owned by an innerPort shall be provided by either

   - a PPortPrototype or
   - a PRPortPrototype.

   Either the shortNames of VariableDataPrototypes are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.

3. For the ModeDeclarationGroupPrototype present in the ModeSwitchInterface of the provided outer PortPrototype, exactly one connection via

   - DelegationSwConnector to a provided inner PortPrototype **or**
   - PassThroughSwConnector to a required outer PortPrototype

   with a compatible ModeDeclarationGroupPrototype in the ModeSwitchInterface of the

   - provided inner PortPrototype **or**
   - required outer PortPrototype

   exists.

   Either the shortNames of ModeDeclarationGroupPrototypes are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.

4. For each ClientServerOperation present in the ClientServerInterface of the provided outer PortPrototype, exactly one connection via

   - DelegationSwConnector to a provided inner PortPrototype **or**
   - PassThroughSwConnector to a required outer PortPrototype

   with a compatible ClientServerOperation in the ClientServerInterface of the

   - provided inner PortPrototype **or**

- required outer PortPrototype

exists.

Either the shortNames of ClientServerOperations are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.

5. For each Trigger present in the TriggerInterface of the provided outer PortPrototype, exactly one connection via

   - DelegationSwConnector to a provided inner PortPrototype **or**
   - PassThroughSwConnector to a required outer PortPrototype

   with a compatible Trigger in the TriggerInterface of the provided

   - inner PortPrototype **or**
   - required outer PortPrototype

   exists.

   Either the shortNames of Triggers are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.

⌋

[constr_1085] Compatibility in the case of a flat ECU extract

Imposition time: CP: IT_EcuExt

⌈PortPrototypes of different

- SenderReceiverInterfaces,
- NvDataInterfaces, and
- ParameterInterfaces

are compatible if and only if for at least one

- VariableDataPrototype or
- ParameterDataPrototype

defined in the context of the

- SenderReceiverInterfaces,
- NvDataInterfaces, and
- ParameterInterfaces

of the RPortPrototype, a compatible

- VariableDataPrototype or

- ParameterDataPrototype

exists in the

- SenderReceiverInterfaces,
- NvDataInterfaces, and
- ParameterInterfaces

of the provided PortPrototype.

The compatibility of PortInterface elements depends on the kind of PortInterface and the swImplPolicy attributes of the PortInterface elements.

Either the shortNames of VariableDataPrototypes and ParameterDataPrototypes are used to identify the pair, or a PortInterfaceMapping defines which differently named PortInterface elements correlate with each other.⌋

[constr_1086] SwConnector between two specific PortPrototypes

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈Each pair of PortPrototypes can only be connected by exactly one SwConnector.⌋

[constr_1087] AssemblySwConnector inside CompositionSwComponentType

Imposition time: CP: IT_CompSwcT

⌈An AssemblySwConnector owned by a specific CompositionSwComponentType shall only connect PortPrototypes of SwComponentPrototypes that are owned by the same CompositionSwComponentType.⌋

[constr_1088] DelegationSwConnector inside CompositionSwComponentType

Imposition time: CP: IT_CompSwcT

⌈A DelegationSwConnector owned by a specific CompositionSwComponentType shall only connect a PortPrototype of a SwComponentPrototype that is owned by the same CompositionSwComponentType that also owns the connected delegation PortPrototype.⌋

[constr_1091] RTEEvents that may reference a WaitPoint

Imposition time: CP: IT_CpgExe

⌈A WaitPoint shall only be referenced from the listed RTEEvents:

- DataReceivedEvent
- DataSendCompletedEvent
- ModeSwitchedAckEvent
- AsynchronousServerCallReturnsEvent

⌋

[constr_1092] Restrictions for the ParameterSwComponentType

Imposition time: CP: IT_CpgExe

⌈The following restrictions exist for a ParameterSwComponentType:

- it shall never aggregate a SwcInternalBehavior and
- the only aggregated PortPrototypes shall be PPortPrototypes of type ParameterInterface.

⌋

[constr_1093] Definition of textual strings

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈An ApplicationPrimitiveDataType of category STRING shall have a swTextProps which determines the arraySizeSemantics and swMaxTextSize.⌋

[constr_1095] Values of nDataSets vs. reliability

Imposition time: CP: IT_RteGen

⌈If the value of nDataSets is greater than 0, the value of reliability shall not be set to errorCorrection.⌋

[constr_1096] SwcModeSwitchEvent and WaitPoint

Imposition time: CP: IT_CpgExe

⌈A RunnableEntity that has a WaitPoint shall not be referenced by a SwcModeSwitchEvent.⌋

[constr_1097] RunnableEntity that has a WaitPoint

Imposition time: CP: IT_RteGen

⌈A RunnableEntity that has a WaitPoint shall not be referenced by an RTEEvent that has a reference in the role disabledMode.⌋

[constr_1098] Mode switch and mode disabling

Imposition time: CP: IT_RteGen

⌈A SwcModeSwitchEvent shall not simultaneously reference to the same ModeDeclaration in both the roles mode and disabledMode.⌋

[constr_1100] Unconnected RPortPrototype typed by a DataInterface

Imposition time: CP: IT_RteGen

⌈For any element in an unconnected RPortPrototype typed by a DataInterface, there shall be a requiredComSpec that defines an initValue.⌋

[constr_1101] Mode-related communication

Imposition time: CP: IT_RteGen

⌈An RPortPrototype typed by ModeSwitchInterface shall not be referenced by more than one SwConnector.⌋

[constr_1102] ApplicationError in the scope of one SwComponentType

Imposition time: CP: IT_CpgExe

⌈If a SwComponentType has PortPrototypes typed by different ClientServerInterfaces with equal shortName and ApplicationErrors defined, then the following condition applies: ApplicationErrors with the same shortName shall have **identical** errorCode **values**.⌋

[constr_1103] NonqueuedReceiverComSpec and enableUpdate

Imposition time: CP: IT_CpgExe

⌈A NonqueuedReceiverComSpec that has the value of attribute enableUpdate set to true may not reference a dataElement that in turn is referenced by a VariableAccess in the role dataReadAccess.⌋

[constr_1104] Trigger communication shall not implement an n:1 pattern

Imposition time: CP: IT_RteGen

⌈An RPortPrototype typed by a TriggerInterface shall not be connected to PortPrototypes typed by TriggerInterfaces such that a given Trigger in the TriggerInterface of the RPortPrototype is connected to more than one compatible (see [constr_1081], [constr_1082], and [constr_1251]) Trigger in the TriggerInterfaces of the connected PortPrototypes.⌋

[constr_1105] Value of arraySize

Imposition time: CP: IT_CpgExe

⌈The value of the attribute arraySize of an ImplementationDataTypeElement owned by an ImplementationDataType or ImplementationDataTypeElement of category ARRAY shall be greater than 0 unless attribute ImplementationDataTypeElement.arraySizeHandling exists and is set to the value inheritedFromArrayElementTypeSize.⌋

[constr_1106] Structure shall have at least one element

Imposition time: CP: IT_CpgExe

⌈An ImplementationDataType or ImplementationDataTypeElement of category STRUCTURE shall own at least one ImplementationDataTypeElement.⌋

[constr_1107] Union shall have at least one element

Imposition time: CP: IT_CpgExe

⌈An ImplementationDataType or ImplementationDataTypeElement of category UNION shall own at least one ImplementationDataTypeElement.⌋

[constr_1108] Existence and value of attribute `ApplicationError.errorCode`

Imposition time: CP: `IT_CpgExe`

⌈Attribute `ApplicationError.errorCode` shall exist and its value shall not exceed the closed interval 1..63.

The following exception applies: **only** if attribute `possibleError` is supposed to represent the error code `E_OK`, the value 0 shall be allowed.⌋

[constr_1109] Mapping of `SwComponentPrototype`s typed by a `SensorActuatorSwComponentType`

Imposition time: CP: `IT_RteGen`

⌈A `SwComponentPrototype` typed by a `SensorActuatorSwComponentType` needs to be mapped and run on exactly that ECU that contains the `HwElement` corresponding to the `HwType` that its `SensorActuatorSwComponentType` refers to if it accesses the hardware via the I/O hardware abstraction layer.⌋

[constr_1126] Compatibility of `DataConstr`s

Imposition time: CP: `IT_RteGen`

⌈The `DataConstr` (e.g. the limits) defined by the type of the providing data element shall be within the constraints defined by the type of the requiring data element.

For client-server communication, the following rules apply:

- For `argument`s with attribute `direction` set to the value `in`, the client shall take the role of the *provider* and the server shall take the role of the *requiring side*.
- For `argument`s with attribute `direction` set to the value `inout` the `DataConstr` shall be equal on both sides.
- For `argument`s with attribute `direction` set to the value `out`, the server shall take the role of the *provider* and the client shall take the role of the *requiring side*.

⌋

[constr_1128] Queue length of `ClientServerOperation`s associated with the same `RunnableEntity`

Imposition time: CP: `IT_CpgExe`

⌈If two or more `OperationInvokedEvent`s reference a single `RunnableEntity` the value of the `ServerComSpec` attribute `queueLength` shall be **identical** for all `ServerComSpec`s owned by `PPortPrototype`s of the enclosing `SwComponentType` that reference one of the `ClientServerOperation`s that are also referenced by the `OperationInvokedEvent`s.⌋

[constr_1129] swImplPolicy and NonqueuedReceiverComSpec

Imposition time: CP: IT_CpgExe

⌈The attribute swImplPolicy of a dataElement referenced by a NonqueuedReceiverComSpec **shall not** be set to the value queued.⌋

[constr_1130] swImplPolicy and QueuedReceiverComSpec

Imposition time: CP: IT_CpgExe

⌈The attribute swImplPolicy of a dataElement referenced by a QueuedReceiverComSpec **shall** be set to the value queued.⌋

[constr_1131] swImplPolicy and NonqueuedSenderComSpec

Imposition time: CP: IT_CpgExe

⌈The attribute swImplPolicy of a dataElement referenced by a NonqueuedSenderComSpec **shall not** be set to the value queued.⌋

[constr_1132] swImplPolicy and QueuedSenderComSpec

Imposition time: CP: IT_CpgExe

⌈The attribute swImplPolicy of a dataElement referenced by a QueuedSenderComSpec **shall** be set to the value queued.⌋

[constr_1134] Allowed structure of TEXTTABLE

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The existence of physConstrs is not allowed and compuInternalToPhys shall exist with compuScales consisting of upperLimit and lowerLimit.⌋

[constr_1135] Limit of vt in BITFIELD_TEXTTABLE

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The separator for splitting the string representing the value is "|" and is therefore forbidden to appear in vt.⌋

[constr_1137] Applicability of ParameterInterface

Imposition time: CP: IT_CpgExe

⌈A PPortPrototype typed by a ParameterInterface can **only** be owned by a ParameterSwComponentType or a CompositionSwComponentType.⌋

[constr_1138] Role SwcServiceDependency.assignedPort and DiagEventDebounceMonitorInternal

Imposition time: CP: IT_RteGen

⌈If a SwcServiceDependency aggregates DiagnosticEventNeeds in the role serviceNeeds, then an assignedPort with attribute role set to the value CallbackGetFaultDetectCounter shall only exist if the monitor implements internal debouncing, i.e concrete subclass DiagEventDebounceMonitorInternal is aggregated in the role DiagnosticEventNeeds.diagEventDebounceAlgorithm.⌋

[constr_1140] Combination of `invalidValue` with the attribute `handleInvalid`

Imposition time: CP: `IT_CpgExe`

⌈The combination of setting the attribute `handleInvalid` of the meta-class `InvalidationPolicy` owned by `SenderReceiverInterface` to value `replace` **and** of setting the value of the attribute `initValue` owned by a corresponding `NonqueuedReceiverComSpec` effectively to the value of the `invalidValue` (owned by a corresponding `SwDataDefProps`) is not supported.⌋

[constr_1141] Applicability of the `scope` attribute

Imposition time: CP: `IT_CpgExe`

⌈The attribute `scope` of meta-class `VariableAccess` shall **only** be applied with respect to the aggregation of `VariableAccess` in the following roles:

- `dataReadAccess`
- `dataWriteAccess`
- `dataSendPoint`
- `dataReceivePointByValue`
- `dataReceivePointByArgument`

⌋

[constr_1142] `category` of `CompuMethod` shall not be extended

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈In contrast to the general rule that `category` can be extended by user-specific values it is **not allowed** to extend the meaning of the attribute `category` of meta-class `CompuMethod`.⌋

[constr_1144] `SensorActuatorSwComponentType`, `EcuAbstractionSwComponentType`, and `ComplexDeviceDriverSwComponentType` may only reference a `HwType`

Imposition time: CP: `IT_CpgExe`

⌈The attribute `sensorActuator` of `SensorActuatorSwComponentType`, the attribute `hardwareElement` of `EcuAbstractionSwComponentType`, and the attribute `hardwareElement` of `ComplexDeviceDriverSwComponentType` may **only** reference a `HwType`. References to other subclasses of `HwDescriptionEntity` are not allowed.⌋

[constr_1146] Applicability of a `symbol` for a `CompuScale` in C code

Imposition time: CP: `IT_CpgExe`

⌈The `symbol` attribute shall only be provided for `CompuScale`s where the `category` of the enclosing `CompuMethod` is one of the following:

- `TEXTTABLE`

- SCALE_LINEAR_AND_TEXTTABLE
- SCALE_RATIONAL_AND_TEXTTABLE
- BITFIELD_TEXTTABLE

⌋

[constr_1147] Standardized values for the attribute category of meta-class PortGroup

Imposition time: CP: IT_CompSwcT

⌈The following values of the attribute category of meta-class PortGroup are reserved by the AUTOSAR standard:

- MODE_MANAGEMENT: This represents the usage of the PortGroup for the purpose of mode management
- PARTIAL_NETWORKING: This represents the usage of the PortGroup for the purpose of partial networking

.⌋

[constr_1148] PortInterfaces of PortPrototypes used to connect to NvBlockSwComponentTypes

Imposition time: CP: IT_RteGen

⌈PortInterfaces of PortPrototypes used to connect to NvBlockSwComponentTypes as well as the PortInterfaces used in the context of NvBlockSwComponentTypes shall **always** set the value of the attribute isService to false.⌋

[constr_1149] PortPrototypes used for NV data management

Imposition time: CP: IT_RteGen

⌈A PortPrototype typed by a ClientServerInterface used for NV data management, i.e. the interaction of ApplicationSwComponentTypes with NvBlockSwComponentTypes, shall be typed by ClientServerInterfaces that are compatible to the particular ClientServerInterfaces derived from [6, MOD General Blueprints]. [constr_1148] applies.

This rule shall be imposed.⌋

[constr_1150] Usage of valueType for PortDefinedArgumentValue

Imposition time: CP: IT_RteGen

⌈The valueType (typically this boils down to integer values used to specify an “id”) associated with PortDefinedArgumentValue shall be of category VALUE or TYPE_REFERENCE. The latter case is only supported if the value of category of the target data type is set to VALUE.⌋

[constr_1151] Applicability of `PortInterfaceMapping`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈A `PortInterfaceMapping` is only applicable and valid for a `SwConnector` if the two `PortPrototype`s which are referenced by the `SwConnector` are typed by the same two `PortInterface`s which are mapped by the `PortInterfaceMapping`.⌋

[constr_1152] `category` of `ApplicationArrayElement` and `AutosarDataType` referenced in the role `type` shall be kept in sync

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The value of `category` of an `ApplicationArrayElement` shall always be identical to the value of `category` of the `AutosarDataType` referenced by the `ApplicationArrayElement`.⌋

[constr_1153] Applicability of compatibility requirements for `CompuScale`s

Imposition time: CP: `IT_RteGen`

⌈Compatibility requirements for `CompuScale`s shall only apply for `CompuScale`s where the `category` of the enclosing `CompuMethod` is one of the following:

- `TEXTTABLE`
- `SCALE_LINEAR_AND_TEXTTABLE`
- `SCALE_RATIONAL_AND_TEXTTABLE`
- `TAB_NOINTP`
- `BITFIELD_TEXTTABLE`
- `LINEAR`
- `RAT_FUNC`
- `IDENTICAL`

⌋

[constr_1154] Compatibility of `CompuScale`s for sender-receiver communication and similar use cases

Imposition time: CP: `IT_RteGen`

⌈For sender-receiver communication and similar use cases, it is required that the set of `CompuScale`s defined in the `CompuMethod` of the provider of the communication (i.e. on the side of the `PPortPrototype`) shall be a subset of the set of `CompuScale`s defined in the `CompuMethod` on the required side (i.e. on the side of the `RPortPrototype`).⌋

[constr_1155] Compatibility of `CompuScale`s for client-server communication

Imposition time: CP: `IT_RteGen`

⌈For client-server communication, the following rules apply:

For argument s of direction IN the CompuScale s defined in the CompuMethod of the client (i.e. on the side of the RPortPrototype) shall be a subset of the set of CompuScale s defined in the CompuMethod supported at the server (i.e. on the side of the PPortPrototype).

For argument s of the direction OUT the set of CompuScale s defined in the CompuMethod of the server (i.e. on the side of the PPortPrototype) shall be a subset of the set of CompuScale s defined in the CompuMethod supported at the client (i.e. on the side of the RPortPrototype).

For argument s of direction INOUT the set of CompuScale s defined in the CompuMethod of server and client shall be identical.⌋

[constr_1156] Relevance of "names" of CompuScale s

Imposition time: CP: IT_RteGen

⌈CompuScale s which contribute to tabular conversion by having a compuConst are compatible **if and only if** the "names" of the compuScale s, (namely shortLabel, vt and symbol, according to the priority rules communicated in [TPS_SWCT_01431]) are equal.

If the scale has no compuConst, "names" of CompuScale s are not relevant for compatibility.⌋

[constr_1158] Applicable category s for attribute ImplementationDataType.swDataDefProps.compuMethod

Imposition time: CP: IT_CpgExe

⌈

	IDENTICAL	LINEAR	SCALE_LINEAR	SCALE_LINEAR_AND_TEXTTABLE	RAT_FUNC	SCALE_RATIONAL_AND_TEXTTABLE	TEXTTABLE	TAB_NOINTP	BITFIELD_TEXTTABLE
VALUE							x		x
TYPE_REFERENCE							x		x

⌋

[constr_1159] Consistency of `VariableAndParameterInterfaceMapping` with respect to the referenced `DataInterface`s

Imposition time: CP: `IT_RteGen`

⌈Within one `VariableAndParameterInterfaceMapping` all `firstDataPrototype`s shall belong to one and only one `DataInterface` and all `secondDataPrototype`s shall belong to one other and only one other `DataInterface`.⌋

[constr_1161] Applicability of the attribute `Ref.index`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The use of the attribute `Ref.index` is limited to references to the following meta-classes:

- `ApplicationArrayElement`
- Sub-classes of `AbstractImplementationDataTypeElement`.

⌋

[constr_1163] Compatibility of `CompuMethod`s

Imposition time: CP: `IT_RteGen`

⌈Two `CompuMethod` definitions are compatible if and only if all attributes **except**

- `shortName`
- `desc`
- `introduction`
- `longName`
- `adminData`
- `annotation`
- `displayFormat`

are **identical and** the `compuScale`s and `unit`s are compatible.⌋

[constr_1164] Number of `argument`s owned by a `RunnableEntity`

Imposition time: CP: `IT_CpgExe`

⌈If a given `RunnableEntity` owns `RunnableEntityArgument`s in the role `argument`, then the number of these `RunnableEntityArgument`s shall be identical to the number of applicable `portArgValue`s of the `PortAPIOption` that references the `PortPrototype` that in turn is referenced by the `OperationInvokedEvent` that references the `RunnableEntity` **plus** the number of `ArgumentDataPrototype`s aggregated in the role `argument` by the `ClientServerOperation` referenced by said `OperationInvokedEvent`.⌋

[constr_1165] Applicability of `RunnableEntityArgument`

Imposition time: CP: `IT_CpgExe`

⌈The existence of a `RunnableEntityArgument` is limited to `RunnableEntity`s triggered by a `ClientServerOperation`.⌋

[constr_1166] Restrictions of `ModeRequestTypeMap`

Imposition time: CP: `IT_CpgExe`

⌈For every `ModeDeclarationGroup` referenced by a `ModeDeclarationGroupPrototype` used in a `PortPrototype` typed by a `ModeSwitchInterface`, a `ModeRequestTypeMap` shall exist that points to the `ModeDeclarationGroup` and also to an eligible `ImplementationDataType`.

The `ModeRequestTypeMap` shall be aggregated by a `DataTypeMappingSet` which is referenced from the `SwcInternalBehavior` that is owned by the `ApplicationSwComponentType` that also owns the `PortPrototype`.⌋

[constr_1167] `ImplementationDataType`s used as `ModeRequestTypeMap.implementationDataType`

Imposition time: CP: `IT_CpgExe`

⌈The `ImplementationDataType` referenced by a `ModeRequestTypeMap` shall **either** be

- of `category` `VALUE` **or**
- of `category` `TYPE_REFERENCE` that in turn references an `ImplementationDataType` of `category` `VALUE`.

The `baseType` referenced by the `ImplementationDataType` shall have the attribute `BaseTypeDirectDefinition.baseTypeEncoding` set to `NONE`.⌋

[constr_1168] Compatibility of `ImplementationDataType`s used in the `ModeRequestTypeMap`

Imposition time: CP: `IT_RteGen`

⌈Both `ImplementationDataType`s shall fulfill [constr_1167].

In addition to that, the possible numbers used for representing `ModeDeclaration`s on the side of the mode manager shall match the supported range of the `ImplementationDataType` used for representing `ModeDeclaration`s on the side of the mode user (see [constr_1075]).⌋

[constr_1169] Allowed values for `Trigger.swImplPolicy`

Imposition time: CP: `IT_CpgExe`

⌈The **only** allowed values for the attribute `Trigger.swImplPolicy` are either `STANDARD` (in which case the `Trigger` processing does not use a queue) or `QUEUED` (in which case the processing of `Trigger`s uses a queue).⌋

[constr_1172] Allowed values of SwCalibrationAccessEnum for ModeDeclarationGroupPrototype

Imposition time: CP: IT_CpgExe

⌈The only allowed values of swCalibrationAccess aggregated by ModeDeclarationGroupPrototype are

- notAccessible and
- readOnly.

⌋

[constr_1173] Applicability of AutosarParameterRef referencing a VariableDataPrototype

Imposition time: CP: IT_CpgExe

⌈A reference from AutosarParameterRef to VariableDataPrototype is **only** applicable if the AutosarParameterRef is used in the context of SwAxisGrouped.⌋

[constr_1174] PortInterfaces used in the context of CompositionSwComponentTypes cannot refer to AUTOSAR services

Imposition time: CP: IT_CompSwcT

⌈CompositionSwComponentTypes shall not own PortPrototypes typed by PortInterfaces where the attribute isService is set to true.⌋

[constr_1175] Depending on its category, CompuMethod shall refer to a unit

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈As a CompuMethod specifies the conversion between the physical world and the numerical values, it shall refer to a unit unless the CompuMethod's category is one of TEXTTABLE, BITFIELD_TEXTTABLE, or IDENTICAL.⌋

[constr_1176] Compatibility of CompuScales of category LINEAR and RAT_FUNC

Imposition time: CP: IT_RteGen

⌈CompuScales of category LINEAR and RAT_FUNC are considered compatible if they yield the same conversion.⌋

[constr_1177] Allowed targetCategory for SwPointerTargetProps

Imposition time: CP: IT_CpgExe

⌈If the value of attribute targetCategory exists, then it shall be set to one of the following values:

- TYPE_REFERENCE
- FUNCTION_REFERENCE

- VALUE (only applicable if the SwPointerTargetProps.swDataDefProps refers to a SwBaseType where attribute nativeDeclaration is set to the value "void")

⌋

[constr_1178] Existence of attributes of SwDataDefProps in the context of ImplementationDataType

Imposition time: CP: IT_CpgExe

⌈To remove possible sources of ambiguity, SwDataDefProps used in the context of ImplementationDataType can **only have one of**

- baseType
- swPointerTargetProps
- implementationDataType

⌋

[constr_1181] Numerical values used in ModeDeclaration.value and ModeDeclarationGroup.onTransitionValue

Imposition time: CP: IT_CpgExe

⌈The numerical values used to define the value attributes and the onTransitionValue attribute of a ModeDeclarationGroup shall not overlap.⌋

[constr_1182] Allowed values for InternalTriggeringPoint.swImplPolicy

Imposition time: CP: IT_RteGen

⌈The **only** allowed values for the attribute swImplPolicy of meta-class InternalTriggeringPoint are either STANDARD (in which case the processing of the internal triggering does not use a queue) or QUEUED (in which case the processing of internal triggering uses a queue).⌋

[constr_1184] Consistency of rootDataPrototype and base in the context of ApplicationCompositeElementInPortInterfaceInstanceRef

Imposition time: CP: IT_RteGen

⌈The rootDataPrototype referenced by ApplicationCompositeElementInPortInterfaceInstanceRef shall be owned by the applicable subclass of DataInterface referenced in the role base.

This implies that the rootDataPrototype shall be a ParameterDataPrototype if the base is a ParameterInterface. Otherwise, the rootDataPrototype shall be a VariableDataPrototype.⌋

[constr_1185] Consistency of data types in the context of ApplicationCompositeElementInPortInterfaceInstanceRef

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈The definition of attributes contextDataPrototype and targetDataPrototype shall (via the type-prototype pattern) be enclosed in the context of the definition of the data type used to type rootDataPrototype.⌋

[constr_1186] Consistency of data types in the context of ArVariableInImplementationDataInstanceRef

Imposition time: CP: IT_RteGen

⌈The definition of attributes contextDataPrototype and targetDataPrototype shall be enclosed in the context of the definition of the data type used to type rootVariableDataPrototype.⌋

[constr_1187] Compatibility of VariableDataPrototypes or ParameterDataPrototypes typed by composite data types

Imposition time: CP: IT_RteGen

⌈DataPrototypes of ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY are compatible if one of the following conditions evaluates to true:

1. The underlying ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY are identical
2. The underlying ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY fulfill the following condition:
   - They consist of the same number of elements **and**
   - They are composed of compatible AutosarDataTypes (either ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY **OR** ApplicationPrimitiveDataTypes or ImplementationDataTypes of category VALUE, BOOLEAN, or STRING) in *the same order* **and**
   - All attributes match exactly, except for the shortName of the M1 AutosarDataType.
3. In the context of a DataPrototypeMapping, for each ApplicationCompositeElementDataPrototype of the required DataPrototype a SubElementMapping exists such that a ApplicationCompositeDataTypeSubElementRef in the role firstElement or secondElement exists that references the required ApplicationCompositeElementDataPrototype **and** a corresponding ApplicationCompositeDataTypeSubElementRef exists in the **other** role (i.e. secondElement or firstElement) that in turn references an ApplicationCompositeElementDataPrototype of the provided ApplicationCompositeDataType.

4. If and only if the DataPrototype is **not** typed by an ApplicationDataType but by an ImplementationDataType: in the context of a DataPrototypeMapping, for each ImplementationDataTypeElement of the required DataPrototype a SubElementMapping exists such that a ImplementationDataTypeSubElementRef in the role firstElement or secondElement exists that references the required ImplementationDataTypeElement **and** a corresponding ImplementationDataTypeSubElementRef exists in the **other** role (i.e. secondElement or firstElement) that in turn references an ImplementationDataTypeElement of the provided ImplementationDataType.

⌋

[constr_1188] Existence of ReceiverComSpec.replaceWith

Imposition time: CP: IT_CpgExe

⌈The aggregation of VariableAccess in the role ReceiverComSpec.replaceWith shall exist **if and only if at least one of the following conditions is fulfilled**:

- Attribute ReceiverComSpec.handleOutOfRange is set to the value externalReplacement.
- Attribute SenderReceiverInterface.invalidationPolicy.handleInvalid is set to the value externalReplacement.

⌋

[constr_1190] Only one mapping for composite to primitive use case

Imposition time: CP: IT_RteGen

⌈In the case described by [TPS_SWCT_01195] only one subElementMapping shall exist at the enclosing DataPrototypeMapping.⌋

[constr_1191] Value of Limit shall yield a numerical value

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈After all variability is bound, the content obtained from a limit shall yield a numerical value.⌋

[constr_1192] Compatibility of "IDENTICAL" to "RAT_FUNC" or "LINEAR"

Imposition time: CP: IT_RteGen

⌈Similar to [constr_1176], a CompuScale where the category of the enclosing CompuMethod is set to IDENTICAL is considered compatible with a CompuScale where the category of the enclosing CompuMethod is set to RAT_FUNC or LINEAR if the following rule applies:

$int = \frac{N_0+N_1*phys+N_i*phys^i}{D_0+D_1*phys+D_i*phys^i} = phys$⌋

[constr_1193] ModeDeclaration shall be referenced by at least one ModeTransition in the role enteredMode

Imposition time: CP: IT_RteGen

⌈For each ModeDeclaration, at least one ModeTransition shall reference the ModeDeclaration in the role enteredMode.

This constraint shall apply **only** if there is at least one ModeTransition defined in the context of the enclosing ModeDeclarationGroup and it shall **not** apply to the initialMode.⌋

[constr_1194] Identical ModeTransitions

Imposition time: CP: IT_RteGen

⌈Two ModeDeclarationGroups contain identical modeTransitions if and only if

1. For each ModeTransition defined in the context of the mode provider one ModeTransition with the same shortName is defined in the context of the mode user.
2. Each pair of ModeTransitions in both ModeDeclarationGroups identified by their respective shortName have identical targets (in terms of the shortName of the referenced ModeDeclaration) of the references enteredMode and exitedMode.

⌋

[constr_1195] SwcModeSwitchEvent and the definition of ModeTransition

Imposition time: CP: IT_RteGen

⌈For each pair of ModeDeclarations referenced by a SwcModeSwitchEvent with attribute activation set to onTransition a ModeTransition shall be defined in the corresponding direction (i.e. from exitedMode to enteredMode). This constraint shall only apply if the respective ModeDeclarationGroup defines at least one modeTransition.⌋

[constr_1196] Existence of networkRepresentation vs. compositeNetworkRepresentation

Imposition time: CP: IT_CpgExe

⌈If a ReceiverComSpec or SenderComSpec aggregates SwDataDefProps in the role networkRepresentation, it shall **not** aggregate compositeNetworkRepresentation (and vice versa).⌋

[constr_1197] Existence of compositeNetworkRepresentation shall be comprehensive

Imposition time: CP: IT_CpgExe

⌈If at least one compositeNetworkRepresentation exists then for each leaf ApplicationCompositeElementDataPrototype of the affected Application-

CompositeDataType exactly one compositeNetworkRepresentation shall be defined.

For each such compositeNetworkRepresentation, attributes leafElement and networkRepresentation shall exist.⌋

[constr_1200] Queued communication is not applicable for dataElements owned by PRPortPrototype

Imposition time: CP: IT_CpgExe

⌈The swImplPolicy shall not be set to queued for any dataElement owned by a PRPortPrototype.⌋

[constr_1202] Supported connections by AssemblySwConnector between PortPrototypes typed by a SenderReceiverInterface or NvDataInterface

Imposition time: CP: IT_CompSwcT

⌈

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	Yes	Yes
PPortPrototype	Yes	No	Yes
PRPortPrototype	Yes	Yes	Yes

⌋

[constr_1203] Supported connections by DelegationSwConnector between PortPrototypes typed by a SenderReceiverInterface or NvDataInterface

Imposition time: CP: IT_CompSwcT

⌈

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	Yes	No	Yes
PPortPrototype	No	Yes	Yes
PRPortPrototype	Yes	Yes	Yes

⌋

[constr_1204] Supported connections by AssemblySwConnector between PortPrototypes typed by a ClientServerInterface, ModeSwitchInterface, or TriggerInterface

Imposition time: CP: IT_CompSwcT

⌈

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	Yes	Yes
PPortPrototype	Yes	No	No

▽

△

PRPortPrototype	**Yes**	No	No

⌋

[constr_1205] Supported connections by DelegationSwConnector between PortPrototypes typed by a ClientServerInterface, ModeSwitchInterface, or TriggerInterface

Imposition time: CP: IT_CompSwcT

⌈

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	**Yes**	No	No
PPortPrototype	No	**Yes**	No
PRPortPrototype	No	**Yes**	No

⌋

[constr_1209] Mapping of ModeDeclarations of mode user to ModeDeclaration of mode manager

Imposition time: CP: IT_RteGen

⌈A configuration that maps **several** ModeDeclarations representing modes of a mode user to **one** ModeDeclaration representing a mode of a mode manager shall be rejected.⌋

[constr_1210] Mapping of ModeDeclarations of mode user to all ModeDeclarations of mode manager

Imposition time: CP: IT_RteGen

⌈If a ModeDeclarationMapping exists that references a ModeDeclaration representing a mode of the mode manager, then ModeDeclarationMappings shall exist that map all modes of the mode manager to modes of the mode user.⌋

[constr_1219] Invalidation depends on the value of swImplPolicy

Imposition time: CP: IT_CpgExe

⌈If the value of swImplPolicy of a SenderReceiverInterface.dataElement is set to the value SwImplPolicyEnum.queued, then the enclosing SenderReceiverInterface shall not aggregate in the role invalidationPolicy an InvalidationPolicy

- that references the dataElement and
- where the value of InvalidationPolicy.handleInvalid is set to any value other than HandleInvalidEnum.dontInvalidate.

⌋

[constr_1220] Compatibility of SwBaseType

Imposition time: CP: IT_RteGen

⌈Two SwBaseTypes are compatible if and only if attributes

- baseTypeSize respectively
- byteOrder,
- memAlignment,
- baseTypeEncoding, and
- nativeDeclaration

have identical values.⌋

[constr_1221] DataPrototype is typed by an ApplicationPrimitiveDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If a DataPrototype is typed by an ApplicationPrimitiveDataType, its initValue shall be provided by an ApplicationValueSpecification.

If the underlying ApplicationPrimitiveDataType represents an enumeration, the value provided shall match one of the applicable text values (vt, shortLabel, symbol) defined by the applicable CompuScales.⌋

[constr_1222] category of an AutosarDataType used to type a DataPrototype is set to STRING

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the category of an AutosarDataType used to type a DataPrototype is set to STRING, the ApplicationValueSpecification used to initialize the DataPrototype shall be of category STRING.⌋

[constr_1223] DataPrototype is typed by an ApplicationRecordDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If a DataPrototype is typed by an ApplicationRecordDataType, the corresponding initValue shall be provided by a RecordValueSpecification.⌋

[constr_1224] DataPrototype is typed by an ApplicationArrayDataType

Imposition time: CP: IT_CpgExe

⌈If a DataPrototype is typed by an ApplicationArrayDataType, the corresponding initValue shall be provided by an ArrayValueSpecification (that may contain an ApplicationRuleBasedValueSpecification).⌋

[constr_1225] DataPrototype is typed by an ImplementationDataType that references a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE

Imposition time: CP: IT_CpgExe

⌈If a DataPrototype is typed by an ImplementationDataType that references a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE the applicable ValueSpecification shall be a TextValueSpecification.

In this case the value provided shall match one of the applicable text values (vt, shortLabel, symbol) defined by the applicable CompuScales.⌋

[constr_1226] Applicable range for attribute ExecutableEntityActivationReason.bitPosition

Imposition time: CP: IT_CpgExe

⌈The value of attribute ExecutableEntityActivationReason.bitPosition shall be in the range of 0 .. 31.⌋

[constr_1227] Value of attribute ExecutableEntityActivationReason.bitPosition shall be unique

Imposition time: CP: IT_CpgExe

⌈The value of attributes ExecutableEntityActivationReason.bitPosition and ExecutableEntityActivationReason.symbol shall be unique in the context of the enclosing RunnableEntity.⌋

[constr_1228] RTEEvent that is referenced by a WaitPoint in the role trigger shall not reference ExecutableEntityActivationReason

Imposition time: CP: IT_RteGen

⌈An RTEEvent that is referenced by a WaitPoint in the role trigger shall not reference ExecutableEntityActivationReason in the role activationReasonRepresentation.⌋

[constr_1229] category of ImplementationDataType boils down to VALUE

Imposition time: CP: IT_CpgExe

⌈An ImplementationDataType qualifies as an Integral Primitive Type if and only if either

- its category is VALUE or TYPE_REFERENCE that eventually boils down to VALUE **or**
- its category is ARRAY **and** it has only one subElement **and** one of the following conditions applies:
  - subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE **and** the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 **and** the baseTypeEncoding is set to NONE.

- subElement.category is set to TYPE_REFERENCE **and** the swDataDefProps.implementationDataType literally represents the Platform Data Type named “uint8”.
- subElement.category is set to TYPE_REFERENCE **and** the attribute swDataDefProps.implementationDataType.shortName is set to “uint8” **and** swDataDefProps.baseType.baseTypeDefinition.nativeDeclaration does not exist.

⌋

[constr_1230] ApplicationDataType that qualifies for Integral Primitive Type

Imposition time: CP: IT_CpgExe

⌈An ApplicationDataType qualifies as an Integral Primitive Type if and only if **all** the following conditions apply:

- ApplicationDataType.category is set to BOOLEAN, VALUE, STRING, or ARRAY
- in the applicable scope a DataTypeMap is available that refers to the given ApplicationDataType
- the found DataTypeMap refers to an ImplementationDataType that fulfills the requirements of [constr_1229]

⌋

[constr_1231] ConsistencyNeeds aggregated by CompositionSwComponentType

Imposition time: CP: IT_CompSwcT

⌈If ConsistencyNeeds are aggregated by a CompositionSwComponentType the associations stereotyped ≪instanceRef≫ may only refer to context and target elements within the context of this CompositionSwComponentType.⌋

[constr_1232] ConsistencyNeeds aggregated by AtomicSwComponentType

Imposition time: CP: IT_CpgExe

⌈If ConsistencyNeeds are aggregated by a AtomicSwComponentType the associations stereotyped ≪instanceRef≫ may only refer to context and target elements within the context of this AtomicSwComponentType.⌋

[constr_1233] InstantiationTimingEventProps shall only reference TimingEvent

Imposition time: CP: IT_RteGen

⌈An InstantiationTimingEventProps shall only reference TimingEvent in the role refinedEvent. A reference to other kinds of RTEEvents is not supported.⌋

[constr_1234] Value of RunnableEntity.symbol

Imposition time: CP: IT_RteGen

⌈The value of a RunnableEntity.symbol owned by an NvBlockSwComponentType that is triggered by an OperationInvokedEvent shall only be taken from the set of API names associated with the NvM.⌋

[constr_1237] Scope of mapped ClientServerOperations in the context of a ClientServerOperationMapping

Imposition time: CP: IT_RteGen

⌈All ClientServerOperations referenced by a ClientServerOperationMapping in the role firstOperation shall belong to exactly one ClientServerInterface.

All ClientServerOperations referenced by a ClientServerOperationMapping in the role secondOperation shall belong to exactly one other ClientServerInterface.⌋

[constr_1238] Scope of mapped ApplicationErrors in the context of a ClientServerOperationMapping

Imposition time: CP: IT_RteGen

⌈All ApplicationErrors referenced by a ClientServerApplicationErrorMapping in the role firstApplicationError shall belong to exactly one ClientServerInterface.

All ApplicationErrors referenced by a ClientServerApplicationErrorMapping in the role secondApplicationError shall belong to exactly one other ClientServerInterface.⌋

[constr_1240] Consistency of ArgumentDataPrototypes within the context of a ClientServerOperationMapping

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈Unless the reference in the role ClientServerOperationMapping.firstToSecondDataTransformation exists, for each argument owned by

- a ClientServerOperationMapping.firstOperation and
- ClientServerOperationMapping.secondOperation,

a reference in the role

- ClientServerOperationMapping.argumentMapping.firstDataPrototype or
- ClientServerOperationMapping.argumentMapping.secondDataPrototype

shall exist, originated by one of the ClientServerOperationMapping.argumentMappings owned by the mentioned ClientServerOperationMapping.⌋

[constr_1241] Compound Primitive Data Types and invalidValue

Imposition time: CP: IT_CpgExe

⌈Compound Primitive Data Types that have set the value of category other than STRING shall **not** define invalidValue.⌋

[constr_1242] Restriction of invalidValue for ApplicationPrimitiveDataType of category STRING

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈invalidValue for ApplicationPrimitiveDataType of category STRING ([constr_1241] applies) is restricted to either a compatible ApplicationValueSpecification or a ConstantReference that in turn points to a compatible ApplicationValueSpecification.⌋

[constr_1243] NumericalOrText shall either define vf or vt

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈Within the context of one NumericalOrText, **either** the attribute vf **or** the attribute vt shall be defined. The existence of both attributes at the same time is not permitted.⌋

[constr_1244] DataPrototypes used in application software shall not be typed by C enums

Imposition time: CP: IT_CpgExe

⌈A ImplementationDataType that is used to type a DataPrototype owned by an AtomicSwComponentType shall not set swDataDefProps.additionalNativeTypeQualifier to enum.⌋

[constr_1245] Consideration of ModeTransitions for the compatibility of ModeDeclarationGroups

Imposition time: CP: IT_RteGen

⌈One of the following conditions for the consideration of ModeTransitions for the compatibility of ModeDeclarationGroups shall apply:

- **Either** the mode provider **or** the mode user define ModeTransitions.
- The ModeTransitions defined in the context of the mode provider are **identical** to the ModeTransitions defined in the context of the mode user **or** a ModeDeclarationMapping mapping is applied.

⌋

[constr_1246] Consistency of firstMode and secondMode in the scope of one ModeDeclarationMappingSet

Imposition time: CP: IT_RteGen

⌈Within the scope of one ModeDeclarationMappingSet,

- all firstModes shall belong to one and only one ModeDeclarationGroup and

- all secondModes shall belong to one and only one **other** ModeDeclarationGroup.

⌋

[constr_1247] Consistency of ModeDeclarationMappingSet with respect to the referenced firstModeGroup and secondModeGroup

Imposition time: CP: IT_RteGen

⌈If a ModeDeclarationGroupPrototypeMapping.modeDeclarationMappingSet exists, then

- the ModeDeclarationGroup owning the modeDeclarations referenced in the role firstMode shall be the type of the ModeDeclarationGroupPrototypeMapping.firstModeGroup and
- the ModeDeclarationGroup owning the modeDeclarations referenced in the role secondMode shall be the type of the ModeDeclarationGroupPrototypeMapping.secondModeGroup.

⌋

[constr_1248] Compatibility of PortPrototypes of different DataInterfaces in the context of a PassThroughSwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different DataInterfaces are considered compatible if and only if

1. For **at least one** VariableDataPrototype or ParameterDataPrototype defined in the context of the DataInterface of the required outer PortPrototype a compatible VariableDataPrototype or ParameterDataPrototype exists in the DataInterface of the provided outer PortPrototype.

   Either the shortName of VariableDataPrototypes and ParameterDataPrototypes are used to identify the pair **or** a PortInterfaceMapping exists that defines which differently named elements of PortInterfaces correlate with each other.

2. For each such pair, the values of the PortInterface.isService attributes are identical.

⌋

[constr_1249] Compatibility of ModeSwitchInterfaces in the context of a PassThroughSwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different ModeSwitchInterfaces are considered compatible if and only if

1. For the ModeDeclarationGroupPrototype defined in the context of the ModeSwitchInterface of the required outer PortPrototype a compatible ModeDeclarationGroupPrototype exists in the ModeSwitchInterface of the provided outer PortPrototype.

   Either the shortNames of the ModeDeclarationGroupPrototypes are used to identify the pair **or** a ModeInterfaceMapping exists that maps the corresponding ModeDeclarationGroupPrototypes.

2. For each such pair, the values of the PortInterface.isService attributes are identical.

⌋

[constr_1250] Compatibility of ClientServerInterfaces in the context of a PassThroughSwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different ClientServerInterfaces are considered compatible if and only if

1. For **at least one** ClientServerOperation defined in the context of the ClientServerInterface of the provided outer PortPrototype a compatible ClientServerOperation exists in the ClientServerInterface of the required outer PortPrototype.

   Either the shortNames of the ClientServerOperations are used to identify the pair **or** a ClientServerInterfaceMapping exists that maps the corresponding ClientServerOperations.

2. For each such pair, the values of the PortInterface.isService attributes are identical.

⌋

[constr_1251] Compatibility of PortPrototypes of TriggerInterfaces in the context of a PassThroughSwConnector

Imposition time: CP: IT_RteGen

⌈PortPrototypes of different TriggerInterfaces are considered compatible if and only if

1. For **at least one** Trigger defined in the context of the TriggerInterface of the required outer PortPrototype a compatible Trigger exists in the TriggerInterface of the provided outer PortPrototype.

   Either the shortName of Triggers are used to identify the pair **or** a TriggerInterfaceMapping exists that refers to one of the Triggers in the role firstTrigger and to the other in the role secondTrigger.

2. For each such pair, the values of the PortInterface.isService attributes are identical.

⌋

[constr_1252] Creation of a loop involving a PassThroughSwConnector is not allowed

Imposition time: CP: IT_CompSwcT

⌈A PassThroughSwConnector is not allowed if the required outer PortPrototype is directly or indirectly connected to the provided outer PortPrototype without the placement of a SwComponentPrototype typed by an AtomicSwComponentType in the chain of SwConnectors.⌋

[constr_1253] Allowed multiplicities for attributes of VariationPointProxy depending on the applicable binding time and the value of VariationPointProxy.category

Imposition time: CP: IT_CpgExe

⌈

BindingTime	category	Allowed Attribute Multiplicity
PreBuild	VALUE	valueAccess [1], implementationDataType [0..1]
	CONDITION	conditionAccess [1]
PostBuild	VALUE	postBuildValueAccess [1], implementationDataType [1]
	CONDITION	postBuildVariantCondition [1..*], conditionAccess [0..1]

⌋

[constr_1254] Definition of a pointer to a pointer

Imposition time: CP: IT_CpgExe

⌈AUTOSAR does **not** support the definition of a pointer to a pointer by defining an ImplementationDataType of category DATA_REFERENCE that aggregates SwDataDefProps in the role swDataDefProps that in turn aggregate SwPointerTargetProps in the role swPointerTargetProps with attribute targetCategory set to DATA_REFERENCE that in turn aggregates SwDataDefProps in the role swDataDefProps that aggregates SwPointerTargetProps in the role swPointerTargetProps that references an ImplementationDataType of category e.g. VALUE.⌋

[constr_1255] ApplicationPrimitiveDataTypes of category BOOLEAN and STRING

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If a Unit is referenced from within SwDataDefProps and/or PhysConstrs owned by ApplicationPrimitiveDataTypes of category BOOLEAN and STRING, this Unit shall represent a meaningless unit, i.e. the referenced physicalDimension shall not define any exponent value other than 0.⌋

[constr_1256] Acknowledgement feedback in n:1 writer case

Imposition time: CP: IT_CpgExe

⌈Within the scope of one SwcInternalBehavior, it is **not** allowed that two or more aggregated RunnableEntitys own either dataSendPoints or dataWriteAccesss that in turn point to the identical accessedVariable.autosarVariable.targetDataPrototype **if** the attribute transmissionAcknowledge exists in the context of the SenderComSpec owned by the dataSendPoint.accessedVariable.autosarVariable.portPrototype (or the respective construct for dataWriteAccess) that also refers to said dataElement.⌋

[constr_1257] No WaitPoints allowed

Imposition time: CP: IT_RteGen

⌈A RunnableEntity referenced by an InitEvent in the role startOnEvent shall not aggregate a.⌋

[constr_1258] Value of minimumStartInterval for RunnableEntitys triggered by an InitEvent

Imposition time: CP: IT_RteGen

⌈The value of the attribute ExecutableEntity.minimumStartInterval for a RunnableEntity that is triggered by an InitEvent shall be set to 0.⌋

[constr_1259] Aggregation of AsynchronousServerCallPoint and AsynchronousServerCallResultPoint

Imposition time: CP: IT_RteGen

⌈A RunnableEntity referenced by an InitEvent in the role startOnEvent may aggregate an AsynchronousServerCallPoint but it shall not aggregate an AsynchronousServerCallResultPoint.⌋

[constr_1260] No mode disabling for InitEvents

Imposition time: CP: IT_RteGen

⌈An InitEvent shall not have a reference to a ModeDeclaration in the role disabledMode.⌋

[constr_1263] Existence of ModeErrorBehavior.defaultMode

Imposition time: CP: IT_RteGen

⌈The optional attribute ModeErrorBehavior.defaultMode **shall exist** if the value of the attribute ModeErrorBehavior.errorReactionPolicy is set to defaultMode.⌋

[constr_1268] `ArgumentDataPrototype.direction` shall be preserved in a `ClientServerOperationMapping`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈Within the context of a `ClientServerOperationMapping`, the value of the argument `ArgumentDataPrototype.direction` of two mapped `ArgumentDataPrototype` shall be identical.⌋

[constr_1269] Number of `argument`s shall be preserved in a `ClientServerOperationMapping`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈Within the context of a `ClientServerOperationMapping`, the number of `argument`s of `firstOperation` and `secondOperation` shall be identical.⌋

[constr_1270] `ArgumentDataPrototype` shall be mapped only once in a `ClientServerOperationMapping`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈Within the context of a `ClientServerOperationMapping`, each `argument` shall only be referenced **once** in the role `firstDataPrototype` or `secondDataPrototype`.⌋

[constr_1271] `RecordValueSpecification.field`s shall be identical to the number of `ApplicationRecordDataType.element`s

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The initialization of a `DataPrototype` typed by an `ApplicationRecordDataType` through a `RecordValueSpecification` shall exactly match the structure of the `ApplicationRecordDataType`.

For this reason, it is required that the number of `RecordValueSpecification.field`s shall be identical to the number of `ApplicationRecordDataType.element`s.⌋

[constr_1272] `RecordValueSpecification.field`s shall be identical to the number of `subElement`s of `ImplementationDataType` of `category` `STRUCTURE`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The initialization of an `DataPrototype` typed by an `ImplementationDataType` of `category` `STRUCTURE` through a `RecordValueSpecification` shall exactly match the structure of the `ImplementationDataType` of `category` `STRUCTURE`.

For this reason, it is required that the number of `RecordValueSpecification.field`s shall be identical to the number of `ImplementationDataType.subElement`s.⌋

[constr_1273] Rules for the initialization of ApplicationArrayDataType through ArrayValueSpecification

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The following rules apply for the initialization of a DataPrototype typed by an ApplicationArrayDataType through an ArrayValueSpecification:

- If the primitive attribute ApplicationArrayDataType.element.arraySizeSemantics is set to **fixedSize** then the ArrayValueSpecification shall exactly match the structure of the ApplicationArrayDataType.

  This means that the number of ArrayValueSpecification.elements shall be identical to the value of ApplicationArrayDataType.element.maxNumberOfElements.

- If the primitive attribute ApplicationArrayDataType.element.arraySizeSemantics is set to **variableSize** and the ArrayValueSpecification **does not define** attribute intendedPartialInitializationCount then ArrayValueSpecification shall **exactly** match the structure of the ApplicationArrayDataType.

  This means that the number of ArrayValueSpecification.elements shall be identical to the value of ApplicationArrayDataType.element.maxNumberOfElements.

- If the primitive attribute ApplicationArrayDataType.element.arraySizeSemantics is set to **variableSize** and the ArrayValueSpecification specifies a value for attribute intendedPartialInitializationCount then ArrayValueSpecification shall contain **exactly** intendedPartialInitializationCount elements.

  This includes the case that the value of intendedPartialInitializationCount is set to 0 (i.e. “empty” initialization) and the case that the intendedPartialInitializationCount is set to the value of the respective ApplicationArrayElement.maxNumberOfElements (i.e. “full” initialization).

⌋

[constr_1274] Rules for the initialization of array-shaped ImplementationDataType with a fixed size through ArrayValueSpecification

Imposition time: CP: IT_CpgExe

⌈The following rule applies for the initialization of a DataPrototype typed by an ImplementationDataType of category ARRAY where attribute ImplementationDataType.subElement.arraySizeSemantics is set to fixedSize through an ArrayValueSpecification: the ArrayValueSpecification shall exactly match the structure of the ImplementationDataType.

This means that the number of ArrayValueSpecification.elements shall be identical to the value of ImplementationDataType.subElement.arraySize.⌋

[constr_1277] SwDataDefProps.swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePointByValue

Imposition time: CP: IT_CpgExe

⌈The SwDataDefProps.swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePointByValue shall not be set to queued.⌋

[constr_1278] PhysConstrs references a Unit

Imposition time: CP: IT_RteGen

⌈DataConstrs are only compatible if the DataConstr.dataConstrRule.physConstrs.unit are compatible or neither DataConstr.dataConstrRule.physConstrs.unit exist.⌋

[constr_1279] Unmapped elements of ApplicationCompositeDataTypes or ImplementationDataTypes and the attribute swImplPolicy

Imposition time: CP: IT_RteGen

⌈If the attribute swImplPolicy is set to queued, then it is not allowed to have unmapped elements of ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY on the "target" end.⌋

[constr_1280] Unmapped dataElement on the "target" end shall have an initValue

Imposition time: CP: IT_RteGen

⌈If elements of ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY are not considered in a SubElementMapping **and** the NonqueuedReceiverComSpec is aggregated by an AbstractRequiredPortPrototype referenced by the "target" end, then the enclosing dataElement shall have an initValue.⌋

[constr_1282] Restriction on the use of RuleBasedValueSpecification or a ReferenceValueSpecification for the specification of an invalidValue

Imposition time: CP: IT_CpgExe

⌈The aggregation of a RuleBasedValueSpecification or a ReferenceValueSpecification for the definition of a ApplicationPrimitiveDataType.swDataDefProps.invalidValue is not supported.⌋

[constr_1284] Limitation of the use of TextValueSpecification

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈TextValueSpecification shall **only** be used in the context of an AutosarDataType that references a CompuMethod in the role ImplementationDataType.swDataDefProps.compuMethod of category TEXTTABLE and BITFIELD_TEXTTABLE.⌋

[constr_1285] Applicability of roles vs. PortPrototypes

Imposition time: CP: IT_RteGen

⌈The aggregation of AutosarVariableRef aggregated by NvBlockDataMapping in the roles writtenNvData, writtenReadNvData, or readNvData is subject to limitation, depending on the applicable subclass of PortPrototype:

- The role writtenNvData shall only be used if the corresponding PortPrototype is a RPortPrototype
- The role writtenReadNvData shall only be used if the corresponding PortPrototype is a PRPortPrototype
- The role readNvData shall only be used if the corresponding PortPrototype is a PPortPrototype

⌋

[constr_1286] serverArgumentImplPolicy and ArgumentDataPrototype typed by primitive data types

Imposition time: CP: IT_CpgExe

⌈The value of the attribute ArgumentDataPrototype.serverArgumentImplPolicy shall **not** be set to useVoid for an ArgumentDataPrototype of direction in that is typed by an AutosarDataType that resolves to a primitive C data type (see [TPS_SWCT_01565]).⌋

[constr_1287] Compatibility of SenderReceiverInterfaces with respect to invalidationPolicy

Imposition time: CP: IT_RteGen

⌈VariableDataPrototypes defined in the context of the SenderReceiverInterface are only compatible if the invalidationPolicys have the same value.⌋

[constr_1288] Allowed Attributes vs. `category` for `DataPrototype`s typed by `ImplementationDataType`s

Imposition time: CP: `IT_CpgExe`

⌈

Attributes of SwDataDefProps	Root Element			Attribute Existence per Category						
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
additionalNativeTypeQualifier										
annotation	x	x	*	*	*	*	*	*	*	*
baseType										
compuMethod										
dataConstr.dataConstrRule.physConstrs	x	x		d/c[12]			d/c			d/c
dataConstr.dataConstrRule.internalConstrs	x	x		0..1			0..1			0..1
displayFormat	x	x		0..1			0..1	0..1	0..1	0..1
displayPresentation	x	x		0..1			0..1			0..1
implementationDataType										
invalidValue										
stepSize	x	x		0..1						0..1
swAddrMethod	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swBitRepresentation										
swCalibrationAccess	x	x		0..1			0..1	0..1	0..1	0..1
swCalprmAxisSet										
swComparisonVariable										
swDataDependency										
swHostVariable										
swImplPolicy	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution										
swInterpolationMethod										
swIsVirtual										
swPointerTargetProps										
swPointerTargetProps.swDataDefProps										
swPointerTargetProps.functionPointerSignature										
swRecordLayout										
swRefreshTiming	x	x		0..1			0..1	0..1	0..1	0..1
swTextProps										
swValueBlockSize										

▽

[12] don't care

△

Attributes of SwDataDefProps	Root Element			Attribute Existence per Category						
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
swValueBlockSizeMult										
unit										
valueAxisDataType										

⌋

[constr_1289] Allowed Attributes vs. category for DataPrototypes typed by ApplicationDataTypes

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

Attributes of SwDataDefProps	Root El.			Attribute Existence per Category											
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
additionalNativeTypeQualifier																
annotation	x	x	x	*	*	*	*	*	*	*	*	*	*	*	*	*
baseType																
compuMethod																
dataConstr.dataConstrRule.physConstrs	x	x		0..1	0..1		0..1		0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.internalConstrs	x	x		d/c[13]	d/c		d/c		d/c			d/c	d/c	d/c	d/c	d/c
displayFormat	x	x		0..1	0..1		0..1	0..1	0..1			0..1	0..1	0..1	0..1	0..1
displayPresentation	x	x		0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
implementationDataType																
invalidValue																
stepSize	x	x	x	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAddrMethod	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1

▽

[13] don't care

△

Attributes of SwDataDefProps	Root El.			Attribute Existence per Category												
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
swAlignment	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swBitRepresentation																
swCalibrationAccess	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet																
swCalprmAxisSet.swCalprmAxis/SwAxisGrouped.swCalprmRef		x	x				0..1					0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.swVariableRef		x	x				0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet.swCalprmAxis/SwAxisGrouped.sharedAxisType																
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.inputVariableType																
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.unit																
swComparisonVariable			x									0..1	0..1	0..1	0..1	0..1
swDataDependency	x	x		0..1					0..1			0..1	0..1	0..1	0..1	0..1
swHostVariable																
swImplPolicy	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution																
swInterpolationMethod	x	x	x	0..1						0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIsVirtual	x	x		0..1					0..1			0..1	0..1	0..1	0..1	0..1
swPointerTargetProps																
swRecordLayout																
swRefreshTiming	x	x		0..1	0..1			0..1	0..1							
swTextProps																
swValueBlockSize																
swValueBlockSizeMult																
unit																
valueAxisDataType																

⌋

[constr_1290] Limitation on the number of PPortComSpecs in the context of one PPortPrototype

Imposition time: CP: IT_CpgExe

⌈Within the context of one PPortPrototype, there can only be **one** (sub-class of) PPortComSpec that references a given

- dataElement (in the case of SenderComSpec),
- operation (in the case of ServerComSpec),

- modeGroup (in the case of ModeSwitchSenderComSpec),
- variable (in the case of NvProvideComSpec), or
- parameter (in the case of ParameterProvideComSpec).

⌋

[constr_1291] Limitation on the number of RPortComSpecs in the context of one RPortPrototype

Imposition time: CP: IT_CpgExe

⌈Within the context of one RPortPrototype, there can only be **one** RPortComSpec that references a given

- dataElement (in the case of ReceiverComSpec),
- operation (in the case of ClientComSpec),
- modeGroup (in the case of ModeSwitchReceiverComSpec),
- variable (in the case of NvRequireComSpec), or
- parameter (in the case of ParameterRequireComSpec).

⌋

[constr_1292] Limitation on the number of RPortComSpecs/PPortComSpecs in the context of one PRPortPrototype

Imposition time: CP: IT_CpgExe

⌈Within the context of one PRPortPrototype, there can only be **one** RPortComSpec and **one** PPortComSpec that references a given

- dataElement (in the case of ReceiverComSpec/SenderComSpec),
- operation (in the case of ClientComSpec/ServerComSpec),
- modeGroup (in the case of ModeSwitchReceiverComSpec/ModeSwitch-SenderComSpec), or
- variable (in the case of NvRequireComSpec/NvProvideComSpec).

⌋

[constr_1295] PortInterfaces and category DATA_REFERENCE

Imposition time: CP: IT_CpgExe

⌈A DataPrototype defined in the context of a PortInterface used by an

- ApplicationSwComponentType or
- SensorActuatorSwComponentType

that is (after potential indirections via TYPE_REFERENCE are resolved) either typed by or mapped to an ImplementationDataType of category DATA_REFERENCE shall only be used if either the provider or the requester of the information represents

- a ServiceSwComponentType,
- a ComplexDeviceDriverSwComponentType,
- a ParameterSwComponentType,
- an NvBlockSwComponentType, or
- an EcuAbstractionSwComponentType.

⌋

[constr_1296] DataPrototypes used as explicitInterRunnableVariable or implicitInterRunnableVariable and category DATA_REFERENCE

Imposition time: CP: IT_CpgExe

⌈A VariableDataPrototype shall not be aggregated by SwcInternalBehavior in either the role:

- explicitInterRunnableVariable, or
- implicitInterRunnableVariable

if the VariableDataPrototype (after potential indirections via TYPE_REFERENCE are resolved) is either typed by, or mapped to, an:

- ImplementationDataType of category DATA_REFERENCE, or
- ImplementationDataType that contains subElements that (after potential indirections via TYPE_REFERENCE are resolved) are of category DATA_REFERENCE.

⌋

[constr_1298] Existence of attributes if category of a ModeDeclarationGroup is set to EXPLICIT_ORDER

Imposition time: CP: IT_CpgExe

⌈The attributes ModeDeclarationGroup.onTransitionValue and ModeDeclaration.value (for each ModeDeclaration) shall be set if the category of a ModeDeclarationGroup is set to EXPLICIT_ORDER.⌋

[constr_1299] Existence of attributes if category of a ModeDeclarationGroup is set to other than EXPLICIT_ORDER

Imposition time: CP: IT_CpgExe

⌈The attributes ModeDeclarationGroup.onTransitionValue or ModeDeclaration.value (for any ModeDeclaration) shall **not** be set if the category of a ModeDeclarationGroup is set to any value **other than** EXPLICIT_ORDER.⌋

[constr_1300] Primitive DataPrototype on the "source" end shall not be mapped to element of a composite data type on the "target" end of the SwConnector

Imposition time: CP: IT_RteGen

⌈The usage of DataPrototypeMapping or SubElementMapping does not support the following configuration:

- The AutosarDataPrototype contained in the PortPrototype on the "source" end of an SwConnector is typed by an ApplicationPrimitiveDataType of category VALUE or ImplementationDataType of category VALUE or category TYPE_REFERENCE that eventually resolves to category VALUE.
- The DataPrototypeMapping aggregates a subElementMapping that refers to a ImplementationDataTypeElement or ApplicationCompositeElementDataPrototype contained in the PortPrototype on the "target" end.

⌋

[constr_1301] Existence of RoleBasedDataTypeAssignment.role vs. RoleBasedDataAssignment.role

Imposition time: CP: IT_RteGen

⌈The usage of a RoleBasedDataTypeAssignment with attribute role set to the value temporaryRamBlock is only allowed if **no** RoleBasedDataAssignment defined with attribute role set to value defaultValue exists in the owning SwcServiceDependency.⌋

[constr_1302] Restriction of data invalidation

Imposition time: CP: IT_CpgExe

⌈Data invalidation is only applicable in the following cases on the **receiving** side:

1. VariableDataPrototypes typed by either an ApplicationPrimitiveDataType or an ImplementationDataType of category VALUE or TYPE_REFERENCE that boils down to category VALUE that have defined an invalidValue.
2. VariableDataPrototypes typed by either an ApplicationCompositeDataType or an ImplementationDataType of category STRUCTURE or ARRAY, or of category TYPE_REFERENCE that boils down to category STRUCTURE or ARRAY, that have **at least one** primitive element with an invalidValue.

⌋

[constr_1303] Applicability of `TextTableMapping` depending on the value of `CompuMethod.category`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈If a `DataPrototypeMapping` aggregates a `TextTableMapping` then only certain combinations of the value of the applicable `CompuMethod.category` are supported:

- `category` of `firstDataPrototype`: `TEXTTABLE`,
  `category` of `secondDataPrototype`: `TEXTTABLE`
- `category` of `firstDataPrototype`: `SCALE_LINEAR_AND_TEXTTABLE`,
  `category` of `secondDataPrototype`: `TEXTTABLE`
- `category` of `firstDataPrototype`: `TEXTTABLE`,
  `category` of `secondDataPrototype`: `SCALE_LINEAR_AND_TEXTTABLE`
- `category` of `firstDataPrototype`: `BITFIELD_TEXTTABLE`,
  `category` of `secondDataPrototype`: `TEXTTABLE`
- `category` of `firstDataPrototype`: `TEXTTABLE`,
  `category` of `secondDataPrototype`: `BITFIELD_TEXTTABLE`
- `category` of `firstDataPrototype`: `BITFIELD_TEXTTABLE`,
  `category` of `secondDataPrototype`: `BITFIELD_TEXTTABLE`

⌋

[constr_1304] Existence of attribute `bitfieldTextTableMaskFirst`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈The attribute `bitfieldTextTableMaskFirst` shall be defined **only if** the `firstDataPrototype` of a `DataPrototypeMapping` refers to a `CompuMethod` that has the value of `category` set to `BITFIELD_TEXTTABLE`.⌋

[constr_1305] Existence of attribute `bitfieldTextTableMaskSecond`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈The attribute `bitfieldTextTableMaskSecond` shall be defined **only if** the `secondDataPrototype` of a `DataPrototypeMapping` refers to a `CompuMethod` that has the value of `category` set to `BITFIELD_TEXTTABLE`.⌋

[constr_1306] Limitation of `TextTableMapping` for `CompuMethod`s that have the value of `category` set to `BITFIELD_TEXTTABLE`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈For any `TextTableMapping` where both `firstDataPrototype` and `secondDataPrototype` refer to `CompuMethod`s that have the value of `category` set to `BITFIELD_TEXTTABLE` **and** where the attribute `TextTableMapping.valuePair` exists the value of attribute `TextTableMapping.identicalMapping` shall be set to false.⌋

[constr_1307] Consistency of values and masks in `TextTableMapping`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈If a `TextTableMapping` element defines bit masks as `bitfieldTextTableMaskFirst` or `bitfieldTextTableMaskSecond` then all contained `TextTableMapping.valuePair.firstValue`s as well as all `TextTableMapping.valuePair.secondValue`s shall **not** specify a value that would be ruled out when - depending on the given value of `TextTableMapping.mappingDirection` - the relevant bit mask is applied.⌋

[constr_1308] Existence of `NvBlockNeeds.cyclicWritingPeriod`

Imposition time: CP: `IT_RteGen`

⌈The attribute `NvBlockNeeds.cyclicWritingPeriod` shall exist if and only if the attribute `NvBlockNeeds.storeCyclic` exists and its value is set to `true`.⌋

[constr_1309] Existence of `NvBlockDescriptor.timingEvent`

Imposition time: CP: `IT_RteGen`

⌈The attribute `NvBlockDescriptor.timingEvent` shall exist if and only if the `NvBlockDescriptor.nvBlockNeeds.storeCyclic` exists and is set to the value `true`.⌋

[constr_1310] Existence of attributes of meta-class `NvBlockNeeds`

Imposition time: CP: `IT_RteGen`

⌈If in the context of an `ApplicationSwComponentType` the attribute `SwcServiceDependency.serviceNeeds` is implemented by an `NvBlockNeeds` then the following attributes

- `NvBlockNeeds.storeCyclic`
- `NvBlockNeeds.cyclicWritingPeriod`
- `NvBlockNeeds.storeEmergency`
- `NvBlockNeeds.storeImmediate`
- `NvBlockNeeds.storeOnChange`

shall only exist if in the context of the same `SwcServiceDependency` a `SwcServiceDependency.assignedPort` exists that has the attribute `role` set to the value `NvDataPort`.⌋

[constr_1311] Appearance of safety-related possible values of `SwAddrMethod.option`

Imposition time: CP: `IT_RteGen`

⌈Any given collection of values stored in the attribute `SwAddrMethod.option` according to [TPS_SWCT_01456] shall at most include a single value out of the following list:

- **safetyQM**
- **safetyAsilA**
- **safetyAsilB**
- **safetyAsilC**
- **safetyAsilD**

⌋

[constr_1312] PortPrototypes typed by a ParameterInterface

Imposition time: CP: IT_CpgExe

⌈PortPrototypes typed by a ParameterInterface can either be PPortPrototypes or RPortPrototypes. The usage of PRPortPrototypes typed by a ParameterInterface is not supported.⌋

[constr_1313] Completeness of TextTableMapping for the values of a given bit mask on the sender side

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈If a DataPrototypeMapping contains one or more TextTableMapping(s) where the DataPrototype on the **sender side** refers to a CompuMethod of category BITFIELD_TEXTTABLE then all DataPrototypeMapping.textTableMapping shall aggregate a collection of TextTableMapping.valuePair where each possible value of the **sender bit mask**[14] is represented by exactly one TextTableValuePair.firstValue ([TPS_SWCT_01163]) or TextTableValuePair.secondValue ([TPS_SWCT_01164]).⌋

[constr_1314] Profile VSA_LINEAR for ApplicationArrayDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the dynamicArraySizeProfile of ApplicationArrayDataType is set to VSA_LINEAR, the contained ApplicationArrayElement shall fulfill **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

[14] Depending on the applicable case this means either bitfieldTextTableMaskFirst (applies if [TPS_SWCT_01163] is in place) or bitfieldTextTableMaskSecond for the case of [TPS_SWCT_01164].

- The ApplicationArrayElement shall be typed by an ApplicationDataType that is not an ApplicationArrayDataType where the attribute dynamicArraySizeProfile exists.

⌋

[constr_1315] Profile VSA_SQUARE for ApplicationArrayDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the dynamicArraySizeProfile of ApplicationArrayDataType is set to VSA_SQUARE, the contained ApplicationArrayElement shall fulfill **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall not be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value inheritedFromArrayElementTypeSize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

The referred ApplicationArrayDataType shall refer over a chain (under consideration of the number of dimensions of the "root" ApplicationArrayDataType) of nested ApplicationArrayDataTypes with ApplicationArrayElements to an ApplicationDataType that is **not** an ApplicationArrayDataType where the attribute dynamicArraySizeProfile exists.

The last ApplicationArrayDataType in that chain shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling set to the value allIndicesSameArraySize.

All ApplicationArrayDataTypes before shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall not be defined.

- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value inheritedFromArrayElementTypeSize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

⌋

[constr_1316] Profile VSA_RECTANGULAR for ApplicationArrayDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the dynamicArraySizeProfile of ApplicationArrayDataType is set to VSA_RECTANGULAR the contained ApplicationArrayElement shall fulfill **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

The referred ApplicationArrayDataType shall refer over a chain (under consideration of the number of dimensions of the “root” ApplicationArrayDataType) of nested ApplicationArrayDataTypes with ApplicationArrayElements to an ApplicationDataType that is **not** an ApplicationArrayDataType where the attribute dynamicArraySizeProfile exists.

The last ApplicationArrayDataType in that chain shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

All ApplicationArrayDataTypes before shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall set to the value variableSize
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.

- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

⌋

[constr_1317] Profile VSA_FULLY_FLEXIBLE for ApplicationArrayDataType

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the dynamicArraySizeProfile of ApplicationArrayDataType is set to VSA_FULLY_FLEXIBLE, the contained ApplicationArrayElement shall fulfill **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesDifferentArraySize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

The referred ApplicationArrayDataType shall refer over a chain (under consideration of the number of dimensions of the "root" ApplicationArrayDataType) of nested ApplicationArrayDataTypes with ApplicationArrayElements to an ApplicationDataType that is **not** an ApplicationArrayDataType where the attribute dynamicArraySizeProfile exist.

The last ApplicationArrayDataType in that chain shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.
- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

All ApplicationArrayDataTypes before shall have an ApplicationArrayElement that fulfills **all** the following conditions:

- The attribute ApplicationArrayElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ApplicationArrayElement.maxNumberOfElements shall be defined.

- The attribute ApplicationArrayElement.arraySizeHandling shall be set to the value allIndicesDifferentArraySize.
- The ApplicationArrayElement shall be typed by an ApplicationArrayDataType.

⌋

[constr_1318] Profile VSA_LINEAR for ImplementationDataType

Imposition time: CP: IT_CpgExe

⌈If the value of attribute ImplementationDataType.dynamicArraySizeProfile is set to VSA_LINEAR, the ImplementationDataType shall aggregate a VSA Payload ImplementationDataTypeElement that fulfills all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.
- The attribute ImplementationDataTypeElement.category shall be set to ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The VSA Payload ImplementationDataTypeElement shall immediately aggregate another ImplementationDataTypeElement that shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

⌋

[constr_1319] Profile VSA_SQUARE for ImplementationDataType

Imposition time: CP: IT_CpgExe

⌈If the value of attribute ImplementationDataType.dynamicArraySizeProfile is set to VSA_SQUARE, the ImplementationDataType shall aggregate a VSA Payload ImplementationDataTypeElement that fulfills all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The VSA Payload ImplementationDataTypeElement shall immediately aggregate another ImplementationDataTypeElement (representing the first dimension) that shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value inheritedFromArrayElementTypeSize.

All **intermediate** ImplementationDataTypeElements in the aggregation chain that do not terminate the chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value inheritedFromArrayElementTypeSize.

The **terminating** ImplementationDataTypeElement in the aggregation chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

⌋

[constr_1320] Profile VSA_RECTANGULAR for ImplementationDataType

Imposition time: CP: IT_CpgExe

⌈If the value of attribute ImplementationDataType.dynamicArraySizeProfile is set to VSA_RECTANGULAR, the ImplementationDataType shall aggregate a VSA Payload ImplementationDataTypeElement that fulfills all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.
- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The VSA Payload ImplementationDataTypeElement shall immediately aggregate another ImplementationDataTypeElement (representing the first dimension) that shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

All **intermediate** ImplementationDataTypeElements in the aggregation chain that do not terminate the chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

The **terminating** ImplementationDataTypeElement in the aggregation chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

⌋

[constr_1321] Profile VSA_FULLY_FLEXIBLE for ImplementationDataType

Imposition time: CP: IT_CpgExe

⌈If the value of attribute ImplementationDataType.dynamicArraySizeProfile is set to the value VSA_FULLY_FLEXIBLE, the ImplementationDataType shall aggregate a VSA Payload ImplementationDataTypeElement that fulfills all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.
- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The VSA Payload ImplementationDataTypeElement shall immediately aggregate another ImplementationDataTypeElement (representing the first dimension) that shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.category shall be set to STRUCTURE.
- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesDifferentArraySize.

The ImplementationDataTypeElement shall aggregate another ImplementationDataTypeElement that fulfills the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
- The attribute ImplementationDataTypeElement.arraySize shall not be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The **aggregation chain is continued** by a (possibly empty) sequence of a pair of ImplementationDataTypeElements with the following characteristics:

- The first ImplementationDataTypeElement in the pair shall fulfill all the following conditions:
  - The attribute ImplementationDataTypeElement.category shall be set to STRUCTURE.
  - The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
  - The attribute ImplementationDataTypeElement.arraySize shall be defined.
  - The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesDifferentArraySize.
- The second ImplementationDataTypeElement in the pair shall fulfill all the following conditions:
  - The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.
  - The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.
  - The attribute ImplementationDataTypeElement.arraySize shall not be defined.
  - The attribute ImplementationDataTypeElement.arraySizeHandling shall not be defined.

The **terminating** ImplementationDataTypeElement in the aggregation chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.
- The attribute ImplementationDataTypeElement.arraySize shall be defined.
- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

⌋

[constr_1322] Size Indicator for undefined dynamicArraySizeProfile

Imposition time: CP: IT_CpgExe

⌈If the ImplementationDataType.dynamicArraySizeProfile does not exist but the ImplementationDataType is mapped to an ApplicationArrayDataType where the attribute ApplicationArrayDataType.dynamicArraySizeProfile exists, then the ImplementationDataType shall have the category STRUCTURE, representing a Variable-Size Array Data Type with Size Indicator enabled.⌋

[constr_1363] Existence of attributes of DiagnosticValueNeeds

Imposition time: CP: IT_CpgExe

⌈if DiagnosticValueNeeds is aggregated by a SwcServiceDependency in the role serviceNeeds then the attributes

- DiagnosticValueNeeds.diagnosticValueAccess
- DiagnosticValueNeeds.dataLength
- DiagnosticValueNeeds.fixedLength

shall **not** exist.⌋

[constr_1375] Existence of attributes of CompuMethod and related meta-classes depending on the value of the category

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

	Attribute Existence per Category									
Attributes of CompuMethod	IDENTICAL	LINEAR	SCALE_LINEAR	RAT_FUNC	SCALE_RAT_FUNC	TEXTTABLE	BITFIELD_TEXTTABLE	SCALE_LINEAR_AND_TEXTTABLE	SCALE_RATIONAL_AND_TEXTTABLE	TAB_NOINTP
compuInternalToPhys		D(1)	D(1)	D(2)	D(2)	D	D	D(8)	D(2)	D
compuPhysToInternal		D(1)	D(1)	D(2)	D(2)				D(2,3)	
displayFormat	0..1	0..1	0..1	0..1	0..1			0..1	0..1	0..1
Attributes of meta-classes related to CompuMethod										
compuDefaultValue		O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)

▽

△

CompuScale		D/ 1..1	D/ 1..n	D/ 1..1	D/ 1..n	D/ 1..n	D/ 1..n	D/ 1..n	D/ 1..n	D/1..n
CompuScale.compuInverseValue				O(2)	O(2)	O(5)		O(2,5)	O(2,5)	O(5)
CompuScale.lowerLimit		O	D	D(4)	D(4)	D	D	D	D(4)	D
CompuScale.mask							D			
CompuScale.shortLabel						O(7)	O(7)	O(7)	O(7)	
CompuScale.a2lDisplayText						O	O	O	O	
CompuScale.symbol						O(7)	O(7)	O(7)	O(7)	
CompuScale.upperLimit		O	D	D(4)	D(4)	D	D	D	D(4)	D
CompuConst						D/vt	D/vt	D/vt	D/vt	D/vt or vf
CompuRationalCoeffs		D	D	D	D			D	D	
CompuRationalCoeffs.compuDenominator		D/1v	D/1v	D	D			D/1v	D	
CompuRationalCoeffs.compuNumerator		D/2v	D/2v	D	D			D/2v	D	

⌋

[constr_1381] Appearance of core-related possible values of SwAddrMethod.option

Imposition time: CP: IT_RteGen

⌈Any given collection of values stored in the attribute SwAddrMethod.option according to [TPS_SWCT_01456] shall at most include a single value out of the following list:

- **coreGlobal**
- **coreLocal**

⌋

[constr_1382] Mutually exclusive existence of attributes SwVariableRefProxy.autosarVariable vs. SwVariableRefProxy.mcDataInstanceVar

Imposition time: CP: IT_CpgExe

⌈In any given AUTOSAR model, the aggregations SwVariableRefProxy.autosarVariable and SwVariableRefProxy.mcDataInstanceVar shall never exist at the same time.⌋

[constr_1383] Existence of CompuMethod and DataConstr for ImplementationDataTypes of category TYPE_REFERENCE

Imposition time: CP: IT_CpgExe

⌈The existence of ImplementationDataType.swDataDefProps.compuMethod and ImplementationDataType.swDataDefProps.dataConstr for ImplementationDataTypes of category TYPE_REFERENCE is only allowed if the respective ImplementationDataType, after all type references are resolved, is an ImplementationDataType of category VALUE.⌋

[constr_1384] Definition of `invalidValue` for `DataPrototype` typed by `ApplicationPrimitiveDataType` of `category` `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, `RES_AXIS`, and `VAL_BLK`

Imposition time: CP: IT_CpgExe

⌈An invalidValue **shall not** be specified for a DataPrototype typed by ApplicationPrimitiveDataType of category CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, and VAL_BLK.⌋

[constr_1385] `DataPrototype` is typed by an `ImplementationDataType`

Imposition time: CP: IT_CpgExe

⌈If a DataPrototype is typed by an ImplementationDataType, its initValue shall **not** be provided by an ApplicationValueSpecification.⌋

[constr_1386] A `PortDefinedArgumentValue` shall only be defined for `AbstractProvidedPortPrototype`

Imposition time: CP: IT_RteGen

⌈A PortAPIOption which aggregates at least one PortDefinedArgumentValue in the role portArgValue shall reference an AbstractProvidedPortPrototype typed by a ClientServerInterface in the role port.⌋

[constr_1388] `VariationPointProxy` of `category` `VALUE` shall not mix "pre-build" and "post-build" use-cases

Imposition time: CP: IT_CpgExe

⌈If the value of category of the VariationPointProxy is set to VALUE then there can only be one value yielded from the evaluation of a VariationPointProxy. In other words, a VariationPointProxy of category VALUE shall not mix the "pre-build" and "post-build" use-cases.⌋

[constr_1389] Restriction regarding the value of `category` of `VariationPointProxy.implementationDataType`

Imposition time: CP: IT_CpgExe

⌈VariationPointProxy.implementationDataType shall **not** be of category STRUCTURE, ARRAY, UNION, FUNCTION_REFERENCE, and DATA_REFERENCE.

The VariationPointProxy.implementationDataType shall be of category VALUE or TYPE_REFERENCE that, after all references are resolved, yields an ImplementationDataType of category VALUE.⌋

[constr_1390] Restriction to the value of `SenderReceiverInterface.invalidationPolicy.handleInvalid`

Imposition time: CP: IT_CpgExe

⌈If the value of SenderReceiverInterface.invalidationPolicy.handleInvalid is set to any value other than HandleInvalidEnum.dontInvalidate,

then the invalidValue shall not be within the interval defined by the CompuMethod of the applicable dataElement.⌋

[constr_1391] Compatibility of Units in the context of assignment using an ApplicationValueSpecification

Imposition time: CP: IT_CpgExe

⌈If an ApplicationValueSpecification is used in the context of an assignment to an AutosarDataPrototype, then the ApplicationValue-Specification.swValueCont.unit shall be compatible with the Unit used in the definition of the given AutosarDataPrototype, i.e. AutosarDataType.swData-DefProps.unit.⌋

[constr_1392] Compatibility of Units in the context of assignment using an ApplicationRuleBasedValueSpecification

Imposition time: CP: IT_CpgExe

⌈If an ApplicationRuleBasedValueSpecification is used in the context of an assignment to an AutosarDataPrototype then the ApplicationRuleBased-ValueSpecification.swValueCont.unit shall be compatible with the Unit used in the definition of the given AutosarDataPrototype, i.e. AutosarData-Type.swDataDefProps.unit.⌋

[constr_1393] Existence of RuleBasedValueCont.unit

Imposition time: CP: IT_CpgExe

⌈For every RuleBasedValueCont, the reference unit shall exist.⌋

[constr_1395] NvBlockDataMapping shall be complete

Imposition time: CP: IT_RteGen

⌈If an NvBlockDataMapping refers to *sub-elements* or *leaf* elements of the NvData-Interface.nvData in the context of a particular PortPrototype, then **all remaining** *sub-elements* or *leaf* elements **shall effectively be mapped** according to [TPS_SWCT_01659] by means of a collection of NvBlockDataMappings.⌋

[constr_1396] Restriction for the value of attribute category for non-terminating ImplementationDataTypeElements taken to model a Variable-Size Array Data Type

Imposition time: CP: IT_CpgExe

⌈The value of attribute category for non-terminating ImplementationDataType-Elements taken to model a Variable-Size Array Data Type shall **not** be set to TYPE_REFERENCE.⌋

[constr_1397] Existence of attributes of `TransformerHardErrorEvent`

Imposition time: CP: `IT_CpgExe`

⌈For any given `TransformerHardErrorEvent`, **either** the attribute `TransformerHardErrorEvent.operation` **or** `TransformerHardErrorEvent.requiredTrigger` shall exist.⌋

[constr_1398] Existence of attributes of `BaseTypeDirectDefinition`

Imposition time: CP: `IT_CpgExe`

⌈If the value of attribute `BaseTypeDirectDefinition.baseTypeEncoding` is set to `UTF-16` then the attribute `BaseTypeDirectDefinition.byteOrder` shall exist.

The only allowed values of `BaseTypeDirectDefinition.byteOrder` in this case are `mostSignificantByteFirst` and `mostSignificantByteLast`.⌋

[constr_1399] Standardized values of `ModeDeclarationGroup.category`

Imposition time: CP: `IT_CpgExe`

⌈The AUTOSAR standard defines the following values of the attribute `ModeDeclarationGroup.category` with a standardized meaning:

- `EXPLICIT_ORDER`
- `ALPHABETIC_ORDER`

[TPS_SWCT_01010] defines the meaning of these values.

It is **not allowed** to define any custom or project-specific value of the attribute `ModeDeclarationGroup.category`.⌋

[constr_1400] Reference to a specific `DataTransformation`

Imposition time: CP: `IT_RteGen`

⌈A specific `DataTransformation` shall only be referenced by either

- a `DataPrototypeMapping` in the role `firstToSecondDataTransformation` (and potentially `secondToFirstDataTransformation`) **or**
- an `ISignal` in the role `dataTransformation` **or**
- an `ISignalGroup` in the role `comBasedSignalGroupTransformation` **or**
- a `ClientServerOperationMapping` in the role `firstToSecondDataTransformation`

⌋

[constr_1401] Restrictions on the relation between DataPrototypeMapping and DataTransformation

Imposition time: CP: IT_RteGen

⌈A VariableDataPrototype in the context of a PortPrototype shall **not** be referenced by a DataPrototypeMapping that references a DataTransformation while a DataMapping exists that points to this VariableDataPrototype (via the SystemSignal) that also refers to an ISignal that in turn references a DataTransformation.⌋

[constr_1402] Applicability of core-related possible values of SwAddrMethod.option related to SwAddrMethod.sectionInitializationPolicy

Imposition time: CP: IT_CpgExe

⌈If the attribute SwAddrMethod.option is set to coreLocal then the attribute SwAddrMethod.sectionInitializationPolicy of the same SwAddrMethod respectively the MemorySection.swAddrmethod shall be set to either INIT or CLEARED.⌋

[constr_1403] NvBlockDataMappings to a given nvData shall be unambiguous

Imposition time: CP: IT_RteGen

⌈If an NvBlockDataMapping exists that **directly** and **completely** maps a specific NvDataInterface.nvData in the context of a particular PortPrototype, then **no** other NvBlockDataMapping which maps sub-elements of the NvDataInterface.nvData shall exist.⌋

[constr_1404] All NvDataInterface.nvData of PortPrototypes in the context of a specific SwcServiceDependency shall be mapped to the same NvBlockDescriptor

Imposition time: CP: IT_RteGen

⌈In the context of a given SwcServiceDependency (which, in turn, is owned by an AtomicSwComponentType), **all** NvDataInterface.nvData of PortPrototypes referenced by a RoleBasedPortAssignment with attribute RoleBasedPortAssignment.role set to NvDataPort shall be connected (either directly or via the definition of suitable PortInterfaceMappings) to NvDataInterface.nvData (on the side of the NvBlockSwComponentType) that are **completely mapped** (via NvBlockDataMappings) **to the identical** NvBlockDescriptor.ramBlock.⌋

[constr_1407] Definition of SwDataDefProps.dataConstr depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.dataConstr according to [constr_1288] and [constr_1289] is only supported for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY also supports the specification of a SwDataDefProps.dataConstr.⌋

[constr_1408] Definition of SwDataDefProps.displayFormat depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.displayFormat according to [constr_1288] and [constr_1289] is only supported for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY also supports the specification of a SwDataDefProps.displayFormat.⌋

[constr_1409] Definition of SwDataDefProps.dataConstr depending on the capabilities of the element data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.dataConstr according to [constr_1007] and [constr_1009] is only supported for an ApplicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType.element or ImplementationDataType.subElement also supports the specification of a SwDataDefProps.dataConstr.⌋

[constr_1410] Definition of SwDataDefProps.displayFormat depending on the capabilities of the element data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.displayFormat according to [constr_1007] and [constr_1009] is only supported for an ApplicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType.element or ImplementationDataType.subElement also supports the specification of a SwDataDefProps.displayFormat.⌋

[constr_1413] Definition of SwDataDefProps.stepSize depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.stepSize according to [constr_1288] and [constr_1289] is only supported for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY also supports the specification of a SwDataDefProps.stepSize.⌋

[constr_1414] Definition of SwDataDefProps.stepSize depending on the capabilities of the element data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.stepSize according to [constr_1007] and [constr_1009] is only supported for an ApplicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType.element or ImplementationDataType.subElement also supports the specification of a SwDataDefProps.stepSize.⌋

[constr_1415] Supported values of ModeSwitchEventTriggeredActivity.role

Imposition time: CP: IT_RteGen

⌈The only supported value of ModeSwitchEventTriggeredActivity.role is WriteBlock.⌋

[constr_1416] Existence of attribute ApplicationArrayElement.maxNumberOfElements

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The attribute ApplicationArrayElement.maxNumberOfElements shall exist for all ApplicationArrayElements defined in the scope of an ApplicationArrayDataType where the attribute ApplicationArrayDataType.dynamicArraySizeProfile does not exist.⌋

[constr_1417] Invalid connection between NvBlockSwComponentType and other AtomicSwComponentType (I)

Imposition time: CP: IT_RteGen

⌈A configuration where an RPortPrototype owned by an AtomicSwComponentType is simultaneously and directly connected to AbstractProvidedPortPrototypes of a collection of AtomicSwComponentTypes where at least one in the collection is an NvBlockSwComponentType for a matching set of dataElements in all these PortPrototypes shall be considered invalid.⌋

[constr_1418] Invalid connection between NvBlockSwComponentType and other AtomicSwComponentType (II)

Imposition time: CP: IT_RteGen

⌈A configuration where a PRPortPrototype owned by an AtomicSwComponentType is connected to a PPortPrototype owned by an NvBlockSwComponentType for a matching set of dataElements in all these PortPrototypes shall be considered invalid.⌋

[constr_1420] Existence of SwAxisIndividual.inputVariableType

Imposition time: CP: IT_CpgExe

⌈If the reference SwAxisIndividual.inputVariableType does not exist, then either:

- SwAxisIndividual.dataConstr
- SwAxisIndividual.unit

or

- SwAxisIndividual.dataConstr
- SwAxisIndividual.compuMethod.unit

shall exist.⌋

[constr_1422] Value of `category` is `VOID`

Imposition time: CP: IT_CpgExe

⌈If the value of the attribute SwBaseType.category is set to VOID, then the attributes baseTypeSize and baseTypeEncoding shall not exist.⌋

[constr_1423] Completeness of references `ArVariableInImplementationDataInstanceRef.contextDataPrototype`

Imposition time: CP: IT_CpgExe

⌈The reference ArVariableInImplementationDataInstanceRef.contextDataPrototype shall be defined for

- each *leaf* (i.e. the end of a chain of aggregating elements) ImplementationDataTypeElement of category TYPE_REFERENCE in a chain of referencing ImplementationDataTypes which is not the targetDataPrototype
- and each ImplementationDataTypeElement owned by an ImplementationDataType or ImplementationDataTypeElement of category ARRAY in a chain of referencing ImplementationDataTypes

starting from the ImplementationDataTypes of the rootVariableDataPrototype down to the leaf ImplementationDataTypeElement which is typed (directly or indirectly via ImplementationDataType of category TYPE_REFERENCE) by the ImplementationDataType of the targetDataPrototype.⌋

[constr_1424] Existence of `ArVariableInImplementationDataInstanceRef.contextDataPrototype`

Imposition time: CP: IT_CpgExe

⌈The attribute ArVariableInImplementationDataInstanceRef.contextDataPrototype shall only exist for an ImplementationDataTypeElement category TYPE_REFERENCE or ARRAY.⌋

[constr_1425] Definition of `swCalprmAxisSet.swCalprmAxis` / `SwAxisIndividual.swVariableRef` depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe

⌈The definition of a swCalprmAxisSet.swCalprmAxis / SwAxisIndividual.swVariableRef in the context of an InstantiationDataDefProps or a ParameterAccess is only supported for a DataPrototype of category ARRAY if the data type of the ApplicationArrayElement also supports the specification of a swCalprmAxisSet.swCalprmAxis / SwAxisIndividual.swVariableRef according to [constr_1289].

Multiple ApplicationArrayDataTypes may be nested to express multiple array dimensions.⌋

[constr_1426] Consistency of array sizes for axes and input variable array

Imposition time: CP: IT_CpgExe

⌈The number of array dimensions defined by ApplicationArrayDataTypes and the values of the maxNumberOfElements attributes for the array of elements of category CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, or RES_AXIS shall be **identical** to the number of array dimensions and the corresponding values of the maxNumberOfElements of the VariableDataPrototype referenced by SwAxisIndividual.swVariableRef.autosarVariable.⌋

[constr_1427] Definition of swCalprmAxisSet.swCalprmAxis / SwAxisGrouped.swCalprmRef depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe

⌈The definition of a swCalprmAxisSet.swCalprmAxis / SwAxisGrouped.swCalprmRef in the context of an InstantiationDataDefProps or a ParameterAccess is only supported for a DataPrototype of category ARRAY if the data type of the ApplicationArrayElement also supports the specification of a swCalprmAxisSet.swCalprmAxis/ SwAxisGrouped.swCalprmRef according to [constr_1289].

Thereby, multiple ApplicationArrayDataTypes might be nested to express multiple array dimensions.⌋

[constr_1428] Consistency of array sizes for arrays of elements of category CURVE, MAP, CUBOID, CUBE_4, or CUBE_5 arrays and used group axes arrays

Imposition time: CP: IT_CpgExe

⌈The number of array dimensions defined by ApplicationArrayDataTypes and the values of the maxNumberOfElements attributes for the array of elements of category CURVE, MAP, CUBOID, CUBE_4, or CUBE_5 shall be identical to the number of array dimensions and the corresponding values of the maxNumberOfElements of the DataPrototype referenced by SwAxisGrouped.swCalprmRef.arParameter.⌋

[constr_1429] Access to data within PortPrototypes from within RunnableEntitys

Imposition time: CP: IT_CpgExe

⌈For a VariableAccess that is aggregated in the roles

- RunnableEntity.dataWriteAccess
- RunnableEntity.dataReadAccess
- RunnableEntity.dataSendPoint
- RunnableEntity.dataReceivePointByArgument
- RunnableEntity.dataReceivePointByValue

the existence of the following attributes is not allowed:

- VariableAccess.accessedVariable.autosarVariable.contextDataPrototype
- VariableAccess.accessedVariable.autosarVariable.rootVariableDataPrototype
- VariableAccess.accessedVariable.autosarVariableInImplDatatype
- VariableAccess.accessedVariable.localVariable

In other words: in this case, only the references

- VariableAccess.accessedVariable.autosarVariable.portPrototype and
- VariableAccess.accessedVariable.autosarVariable.targetDataPrototype

shall exist and the latter shall **exclusively** refer to a VariableDataPrototype that is aggregated as either

- SenderReceiverInterface.dataElement or
- NvDataInterface.nvData.

⌋

[constr_1430] Access to local data from within RunnableEntitys

Imposition time: CP: IT_CpgExe

⌈For a VariableAccess that is aggregated in the roles

- RunnableEntity.writtenLocalVariable
- RunnableEntity.readLocalVariable

the existence of the following attributes is not allowed:

- VariableAccess.accessedVariable.autosarVariableInImplDatatype
- VariableAccess.accessedVariable.autosarVariable

In other words, **only** the reference VariableAccess.accessedVariable.localVariable shall be used in this case.⌋

[constr_1431] Access to parameters from within RunnableEntitys

Imposition time: CP: IT_CpgExe

⌈For a ParameterAccess that is aggregated in the role RunnableEntity.parameterAccess the existence of the following attributes is not allowed:

- ParameterAccess.accessedParameter.autosarParameter.contextDataPrototype

- ParameterAccess.accessedParameter.autosarParameter.root-ParameterDataPrototype

In other words: in this case, **one** of the following alternatives is allowed to exist:

- a combination of
  - ParameterAccess.accessedParameter.autosarParameter.port-Prototype and
  - ParameterAccess.accessedParameter.autosar-Parameter.targetDataPrototype that **exclusively** refers to a ParameterDataPrototype aggregated by a ParameterInterface in the role parameter.
- ParameterAccess.accessedParameter.localParameter that refers to a ParameterDataPrototype that is either aggregated as
  - InternalBehavior.constantMemory or
  - SwcInternalBehavior.perInstanceParameter or
  - SwcInternalBehavior.sharedParameter.

⌋

[constr_1432] Multiplicity of CommunicationBufferLocking

Imposition time: CP: IT_RteGen

⌈In a concrete aggregated set of PortAPIOption.supportedFeature, CommunicationBufferLocking shall exist.⌋

[constr_1434] CompuScales shall not have identical CompuScale Value Symbolic Names

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈In a CompuMethod that is subject to [constr_1146], no two CompuScales shall have identical CompuScale Value Symbolic Names (according to [TPS_SWCT_01696]).⌋

[constr_1438] ApplicationArrayElement.indexDataType needs to refer to a CompuMethod of category TEXTTABLE

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The reference ApplicationArrayElement.indexDataType shall only point to an ApplicationPrimitiveDataType that in turn refers to a CompuMethod of category TEXTTABLE.⌋

[constr_1439] Requirements on `ApplicationArrayElement` if attribute `indexDataType` exists

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈If `ApplicationArrayElement.indexDataType` exists then the attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `fixedSize` and attribute `arraySizeHandling` shall not exist.⌋

[constr_1440] Size of the `CompuMethod` of `category` `TEXTTABLE` referenced by `ApplicationArrayElement.indexDataType`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The interval defined by the `CompuScale`s contained in the `CompuMethod` referenced by `ApplicationArrayElement.indexDataType` shall start at 0 and include all integer values until `ApplicationArrayElement.maxNumberOfElements` - 1.⌋

[constr_1442] `category` `TYPE_REFERENCE` shall not be used for modeling the "payload" of a `Wrapped Union Data Type`

Imposition time: CP: `IT_CpgExe`

⌈For the modeling of the "payload" part of a `Wrapped Union Data Type` it shall not be possible to use an `ImplementationDataTypeElement` of `category` `TYPE_REFERENCE` that finally (i.e. after all possible indirections are resolved) boils down to `category` `UNION`.⌋

[constr_1444] Limited applicability of `Wrapped Union Data Type`

Imposition time: CP: `IT_CpgExe`

⌈There is no support for the use of `Wrapped Union Data Type` in `PortInterfaceMapping`s or Diagnostics.⌋

[constr_1445] Initialization of the `Member Selector` of a `Wrapped Union Data Type`

Imposition time: CP: `IT_CpgExe`

⌈The `initValue` for the `Member Selector` shall **never be set to any value other than 1**.⌋

[constr_1446] No definition of `invalidValue` for a `Wrapped Union Data Type`

Imposition time: CP: `IT_CpgExe`

⌈The definition of an `invalidValue` for a `DataPrototype` typed by a `Wrapped Union Data Type` is not supported.⌋

[constr_1468] Limitation on the number of `SwcExclusiveAreaPolicy`s

Imposition time: CP: `IT_CpgExe`

⌈An `ExclusiveArea` shall be referenced by **at most** one `SwcExclusiveAreaPolicy`.⌋

[constr_1469] Applicability of constraints depending on the existence of a data transformation

Imposition time: CP: IT_RteGen

⌈[constr_1269], [constr_1270], [constr_1268], and [constr_1240] shall **not** apply under the following conditions:

- A reference from the respective ClientServerOperationMapping to a DataTransformation in the role firstToSecondDataTransformation exists.
- The value of the attribute dataTransformationKind of the referenced DataTransformation is set to DataTransformationKindEnum.asymmetricFromByteArray or DataTransformationKindEnum.asymmetricToByteArray.

⌋

[constr_1516] Completeness of references ArParameterInImplementationDataInstanceRef.contextDataPrototype

Imposition time: CP: IT_CpgExe

⌈The reference ArParameterInImplementationDataInstanceRef.contextDataPrototype shall be defined for

- each *leaf* (i.e. the end of a chain of aggregating elements) ImplementationDataTypeElement of category TYPE_REFERENCE in a chain of referencing ImplementationDataTypes which is not the targetDataPrototype
- and each ImplementationDataTypeElement owned by an ImplementationDataType or ImplementationDataTypeElement of category ARRAY in a chain of referencing ImplementationDataTypes

starting from the ImplementationDataTypes of the rootParameterDataPrototype down to the leaf ImplementationDataTypeElement which is typed (directly or indirectly via ImplementationDataType of category TYPE_REFERENCE) by the ImplementationDataType of the targetDataPrototype.⌋

[constr_1517] Existence of ArParameterInImplementationDataInstanceRef.contextDataPrototype

Imposition time: CP: IT_CpgExe

⌈The attribute ArParameterInImplementationDataInstanceRef.contextDataPrototype shall only exist for an ImplementationDataTypeElement category TYPE_REFERENCE or ARRAY.⌋

[constr_1518] Consistency of data types in the context of ArParameterInImplementationDataInstanceRef

Imposition time: CP: IT_RteGen

⌈The definition of attributes contextDataPrototype and targetDataPrototype shall be enclosed in the context of the definition of the data type used to type rootParameterDataPrototype.⌋

[constr_1519] Existence of attributes vs. category of ApplicationValueSpecification

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

	Attribute Existence per Category										
Attribute of ApplicationValueSpecification	VALUE	VAL_BLK	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
swValueCont	D	D	D	D	D	D	D	D	D	D	D
swValueCont.unit	O	O	O	O	O	O	O	O	O	O	O
swValueCont.swValuesPhys	D	D	D	D	D	D	D	D	D	D	D
swValueCont.swArraysize		D			D	D	D	D	D	D	D
swAxisCont						D	O(1)	O(1)	O(1)	O(1)	O(1)
swAxisCont.unit						O	O	O	O	O	O
swAxisCont.category						D	D	D	D	D	D
swAxisCont.swAxisIndex						D	D	D	D	D	D
swAxisCont.swArraysize						D	D	D	D	D	D
swAxisCont.swValuesPhys						D	O(1)	O(1)	O(1)	O(1)	O(1)

⌋

[constr_1520] Semantics of ObdRatioServiceNeeds.rateBasedMonitoredEvent

Imposition time: CP: IT_RteGen

⌈In the context of an SwcServiceDependency, each DiagnosticEventNeeds referenced in the role rateBasedMonitoredEvent shall only be referenced by at most a single ObdRatioServiceNeeds.⌋

[constr_1521] Reference from AsynchronousServerCallReturnsEvent to AsynchronousServerCallResultPoint

Imposition time: CP: IT_CpgExe

⌈In the context of a RunnableEntity, a given AsynchronousServerCallResultPoint shall only be referenced by one AsynchronousServerCallReturnsEvent in the role eventSource.⌋

[constr_1523] No mode disabling for `OperationInvokedEvent`s

Imposition time: CP: `IT_RteGen`

⌈An `OperationInvokedEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledMode`.⌋

[constr_1538] Restriction for reference `ReceiverComSpec.dataElement`

Imposition time: CP: `IT_CpgExe`

⌈The reference `ReceiverComSpec.dataElement` **shall not** refer to an `ArgumentDataPrototype` or `ParameterDataPrototype`.⌋

[constr_1539] Restriction for `SenderComSpec.dataElement`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈The reference `SenderComSpec.dataElement` **shall not** refer to an `ArgumentDataPrototype` or `ParameterDataPrototype`.⌋

[constr_1540] Existence of `ClientComSpec.operation`

Imposition time: CP: `IT_CpgExe`

⌈The reference `ClientComSpec.operation` **shall exist** if the `AbstractRequiredPortPrototype` that owns the `ClientComSpec` is typed by a `ClientServerInterface`.⌋

[constr_1541] Existence of `ServerComSpec.operation`

Imposition time: CP: `IT_CpgExe`

⌈The reference `ServerComSpec.operation` **shall exist** if the `AbstractProvidedPortPrototype` that owns the `ServerComSpec` is typed by a `ClientServerInterface`.⌋

[constr_1544] Standardized values and multiplicities for the modeling of `SwAxisGeneric` for the definition of a fix axis

Imposition time: CP: `IT_CpgExe`

⌈

`category` of `swAxisType`	`category` of `SwGenericAxisParamType`	Multiplicity of `swGenericAxisParam`	Multiplicity of `vf`
FIX_AXIS_PAR	OFFSET	1	1
	SHIFT	1	1
FIX_AXIS_PAR_DIST	OFFSET	1	1
	DISTANCE	1	1
FIX_AXIS_PAR_LIST	LIST	1	1..*

⌋

[constr_1545] No initialization for fix axis

Imposition time: CP: IT_CpgExe

⌈An ApplicationValueSpecification taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.⌋

[constr_1583] PortInterfaceMapping for DataPrototype typed by Compound Primitive Data Type

Imposition time: CP: IT_CpgExe

⌈There is one very limited use case to apply PortInterfaceMapping for a DataPrototype typed by a Compound Primitive Data Type: adjustment of the shortName of the DataPrototype. Everything else is **not supported**.⌋

[constr_1592] Definition of SwDataDefProps.displayPresentation depending on the capabilities of the data type

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.displayPresentation according to [constr_1288] and [constr_1289] shall only be applied for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY supports the specification of a SwDataDefProps.displayPresentation.⌋

[constr_1602] Definition of SwDataDefProps.displayPresentation depending on the capabilities of the element

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The definition of a SwDataDefProps.displayPresentation according to [constr_1007] and [constr_1009] is only supported for an ApplicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType.element or ImplementationDataType.subElement also supports the specification of a SwDataDefProps.displayPresentation.⌋

[constr_1607] Only Wrapped Union Data Types in PortInterface

Imposition time: CP: IT_CpgExe

⌈Within the scope of a PortInterface, the usage of a union data type is only supported

- for Wrapped Union Data Types.
- for a PortInterface that is used to type a PortPrototype that does not appear as context in an instanceRef owned by a DataMapping. See also [constr_1441].

⌋

[constr_1608] Existence of rootParameterDataPrototype

Imposition time: CP: IT_CpgExe

⌈The reference rootParameterDataPrototype shall exist if and only if

- AutosarDataType of the autosarParameter is a composite data type and
- targetDataPrototype refers to a DataPrototype inside the rootParameterDataPrototype.

⌋

[constr_1609] Existence of rootVariableDataPrototype

Imposition time: CP: IT_CpgExe

⌈The reference rootVariableDataPrototype shall exist if and only if

- the AutosarDataType of the autosarVariable is a composite data type and
- the targetDataPrototype refers to a DataPrototype inside the rootVariableDataPrototype.

⌋

[constr_1610] Existence of SwDataDefProps.swValueBlockSize and SwDataDefProps.swValueBlockSizeMult

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈Attributes SwDataDefProps.swValueBlockSize and SwDataDefProps.swValueBlockSizeMult shall not exist at the same time in the context of a given SwDataDefProps.⌋

[constr_1611] Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement as opposed to ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement

Imposition time: CP: IT_RteGen

⌈For any given ImplementationDataTypeSubElementRef, either the aggregation

- ImplementationDataTypeSubElementRef.implementationDataTypeElement or
- ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement

⌋

[constr_1622] Value of TimingEvent.offset vs. TimingEvent.period

Imposition time: CP: IT_RteGen

⌈If a value is defined for attribute TimingEvent.offset then this value shall be greater than 0 and less or equal than the value of attribute TimingEvent.period of the respective TimingEvent.⌋

[constr_1631] Applicability of DataPrototypeMapping.secondToFirstDataTransformation

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈The reference to DataTransformation in the role DataPrototypeMapping.secondToFirstDataTransformation shall only exist if reference DataPrototypeMapping.firstToSecondDataTransformation exists and refers to a DataTransformation where attribute dataTransformationKind exists and is **not** set to the value symmetric.⌋

[constr_1632] Restriction for roles firstToSecondDataTransformation and secondToFirstDataTransformation

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈If both the reference firstToSecondDataTransformation and the reference secondToFirstDataTransformation exist in the context of the same DataPrototypeMapping then

- the reference in the role firstToSecondDataTransformation shall refer to a DataTransformation with attribute dataTransformationKind set to asymmetricToByteArray and
- the reference in the role secondToFirstDataTransformation shall refer to a DataTransformation with attribute dataTransformationKind set to asymmetricFromByteArray.

⌋

[constr_1634] Allowed combinations of ApplicationDataType.category vs. CompuMethod.category

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

	IDENTICAL	LINEAR	SCALE_LINEAR	SCALE_LINEAR_AND_TEXTTABLE	RAT_FUNC	SCALE_RATIONAL_AND_TEXTTABLE	TEXTTABLE	TAB_NOINTP	BITFIELD_TEXTTABLE
VALUE	x	x	x	x	x	x	x	x	x
VAL_BLK	x	x	x	x	x	x	x	x	x
BOOLEAN							x		
CURVE	x	x	x	x	x	x	x	x	x
MAP	x	x	x	x	x	x	x	x	x
CUBOID	x	x	x	x	x	x	x	x	x
CUBE_4	x	x	x	x	x	x	x	x	x
CUBE_5	x	x	x	x	x	x	x	x	x

⌋

[constr_1635] Relevance of attribute isOptional

Imposition time: CP: IT_RteGen

⌈If a SubElementMapping is defined for the elements of a structured data type then the attribute isOptional[15] shall either not exist for the firstElement and secondElement or it shall have the identical value for the firstElement and secondElement.⌋

[constr_1636] Mapping of data types that represent an Optional Element Structure

Imposition time: CP: IT_CpgExe

⌈An ApplicationRecordDataType with at least one element where attribute isOptional is set to true shall only be mapped to an ImplementationDataType that fulfills the structural requirements to represent an Optional Element Structure (see [TPS_SWCT_01774]).⌋

[15] this is valid for both ApplicationRecordElement and ImplementationDataTypeElement

[constr_1637] Existence of ImplementationDataTypeElement.isOptional vs. ImplementationDataType.isStructWithOptionalElement

Imposition time: CP: IT_CpgExe

⌈If one ImplementationDataType.subElement sets attribute isOptional to the value true then the enclosing ImplementationDataType shall also set attribute isStructWithOptionalElement to true.⌋

[constr_1638] First ImplementationDataTypeElement of ImplementationDataType that represents an Optional Element Structure

Imposition time: CP: IT_CpgExe

⌈The first ImplementationDataTypeElement of ImplementationDataType that represents an Optional Element Structure, i.e. the availabilityBitfield according to [TPS_SWCT_01774], shall not set attribute isOptional to true.⌋

[constr_1639] ImplementationDataTypeElement with attribute isOptional set to true

Imposition time: CP: IT_CpgExe

⌈An ImplementationDataTypeElement where attribute isOptional is set to true, shall set the value of attribute category to one of the following values:

- VALUE
- TYPE_REFERENCE

⌋

[constr_1640] No use of Optional Element Structure for interaction with the diagnostic stack

Imposition time: CP: IT_RteGen

⌈An SwcServiceDependency that aggregates a diagnostic-related subclass of ServiceNeeds shall not refer to any PortPrototype by means of either a RoleBasedPortAssignment or RoleBasedDataAssignment where the respective PortInterface contains any DataPrototype typed by an Optional Element Structure.⌋

[constr_1662] Compatibility of ApplicationRecordDataType and ImplementationDataType that both represent an Optional Element Structure

Imposition time: CP: IT_CpgExe

⌈An ApplicationRecordDataType that represents an Optional Element Structure shall (after all indirections created by ImplementationDataTypes of category TYPE_REFERENCE are resolved) only be mapped/connected to an ImplementationDataType of category STRUCTURE that represents an Op-

tional Element Structure if corresponding pairs of elements have the same value of the attribute isOptional.⌋

[constr_1679] Existence of attribute RoleBasedDataAssignment.usedDataElement.localVariable for RoleBasedDataAssignment.role = signalBasedDiagnostics

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value signalBasedDiagnostics then the reference RoleBasedDataAssignment.usedDataElement.localVariable shall not exist.⌋

[constr_1680] Existence of attribute RoleBasedDataAssignment.usedDataElement.localVariable for RoleBasedDataAssignment.role = AppModeRequestInterface

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value AppModeRequestInterface, then the reference RoleBasedDataAssignment.usedDataElement.localVariable shall not exist.⌋

[constr_1681] Existence of attribute RoleBasedDataAssignment.usedDataElement.localVariable for RoleBasedDataAssignment.role = VerificationStatus

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value VerificationStatus then the reference RoleBasedDataAssignment.usedDataElement.localVariable shall not exist.⌋

[constr_1694] Allowed target of SwDataDefProps.implementationDataType

Imposition time: CP: IT_CpgExe

⌈The reference SwDataDefProps.implementationDataType shall only refer to an ImplementationDataType. Any other subclass of AbstractImplementationDataType is not supported as a reference target.⌋

[constr_1712] Existence of attribute ArrayValueSpecification.intendedPartialInitializationCount

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈An ArrayValueSpecification where attribute intendedPartialInitializationCount exists shall only be applied for the initialization of an ApplicationArrayDataType where attribute arraySizeSemantics is set to variableSize.⌋

[constr_1713] NvBlockDescriptor.writingStrategy.usedDataElement shall refer to AutosarDataPrototype

Imposition time: CP: IT_RteGen

⌈The reference NvBlockDescriptor.writingStrategy.usedDataElement shall **only** refer to an AutosarDataPrototype.⌋

[constr_1714] AutosarDataPrototype shall only be referenced by a single NvBlockDescriptor.writingStrategy

Imposition time: CP: IT_RteGen

⌈If an AutosarDataPrototype in the context of a PortPrototype is referenced from a NvBlockDescriptor.writingStrategy then this AutosarDataPrototype shall not be referenced from any other NvBlockDescriptor.writingStrategy.⌋

[constr_1715] Possible values of attribute NvBlockDescriptor.writingStrategy.role

Imposition time: CP: IT_RteGen

⌈The attribute NvBlockDescriptor.writingStrategy.role shall only have one of the following values (see [TPS_SWCT_01586]):

- **storeAtShutdown**
- **storeImmediate**
- **storeOnChange**

⌋

[constr_1716] Consistency of the attribute NvBlockDescriptor.writingStrategy.role set to storeAtShutdown

Imposition time: CP: IT_RteGen

⌈The existence of NvBlockDescriptor.writingStrategy where attribute role is set to storeAtShutdown is only supported if NvBlockDescriptor.nvBlockNeeds.storeAtShutdown exists and is set to true.⌋

[constr_1717] Consistency of the attribute NvBlockDescriptor.writingStrategy.role set to storeImmediate

Imposition time: CP: IT_RteGen

⌈The existence of NvBlockDescriptor.writingStrategy where attribute role is set to storeImmediate is only supported if NvBlockDescriptor.nvBlockNeeds.storeImmediate exists and is set to true.⌋

[constr_1718] Inheritance of SwDataDefProps.dataConstr from an array data type to the array elements

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈A SwDataDefProps.dataConstr specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.dataConstr. This also applies for multi-dimensional array data types.⌋

[constr_1719] Inheritance of SwDataDefProps.displayFormat from an array data type to the array elements

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈A SwDataDefProps.displayFormat specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.displayFormat. This also applies for multi-dimensional array data types.⌋

[constr_1720] Inheritance of SwDataDefProps.stepSize from an array data type to the array elements

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈A SwDataDefProps.stepSize specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.stepSize. This also applies for multi-dimensional array data types.⌋

[constr_1724] Usage of attribute ClientServerOperation.diagArgIntegrity

Imposition time: CP: IT_RteGen

⌈With the exception of the context of a ServiceSwComponentType, the attribute ClientServerOperation.diagArgIntegrity shall only have the value true if the ClientServerInterface containing the respective ClientServerOperation is used to type a PPortPrototype that is referenced by a RoleBasedPortAssignment aggregated by a SwcServiceDependency that in turn aggregates DiagnosticRoutineNeeds.⌋

[constr_1726] Ordering of MetaDataItemSet.metaDataItem

Imposition time: CP: IT_CpgExe

⌈The ordering of the elements of MetaDataItemSet.metaDataItem shall be **descending** with respect to the value of MetaDataItem.length, such that the MetaDataItem with the largest value of attribute length is located in the first position and the MetaDataItem with the smallest value of attribute length is located in the last position.⌋

[constr_1735] Limitation of the aggregation of AutosarVariableRef in the context of an NvBlockDataMapping owned by a BulkNvDataDescriptor

Imposition time: CP: IT_RteGen

⌈Any NvBlockDataMapping owned by a BulkNvDataDescriptor shall only aggregate an AutosarVariableRef in the role readNvData and nvRamBlockElement (that in turn refers to the BulkNvDataDescriptor.bulkNvBlock).⌋

[constr_1741] Restriction to explicit sending semantics for the usage of DataServices in the context of a SwcServiceDependency that aggregates DiagnosticValueNeeds that in turn is referenced by a DiagnosticIoControlNeeds

Imposition time: CP: IT_RteGen

⌈A dataElement

- that is referenced by a RoleBasedDataAssignment (where the attribute role is set to signalBasedDiagnostics) owned by a SwcServiceDependency that aggregates DiagnosticValueNeeds that in turn is referenced by a DiagnosticIoControlNeeds
- **shall also be referenced** by a VariableAccess aggregated in the role dataSendPoint by a given RunnableEntity that in turn belongs to the enclosing SwcInternalBehavior.
- **shall not be referenced** by a VariableAccess aggregated in the role dataWriteAccess by a given RunnableEntity that in turn belongs to the enclosing SwcInternalBehavior.

⌋

[constr_1754] Aggregation of NumericalRuleBasedValueSpecification

Imposition time: CP: IT_CpgExe

⌈Each ArrayValueSpecification shall only aggregate at most one NumericalRuleBasedValueSpecification in the role element.

If one NumericalRuleBasedValueSpecification is aggregated then it shall be the only aggregated element, i.e. no further ValueSpecification shall exist in the same aggregation where an NumericalRuleBasedValueSpecification is aggregated.⌋

[constr_1755] Aggregation of `CompositeRuleBasedValueSpecification`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈Each `ArrayValueSpecification` shall only aggregate at most one `CompositeRuleBasedValueSpecification` in the role element.

If one `CompositeRuleBasedValueSpecification` is aggregated then it shall be the only aggregated element, i.e. no further `ValueSpecification` shall exist in the same aggregation where an `CompositeRuleBasedValueSpecification` is aggregated.⌋

[constr_1771] Existence of `SwValueCont.unit`

Imposition time: CP: `IT_CpgExe`

⌈For every `SwValueCont`, the reference `unit` shall exist⌋

[constr_1773] Value of attribute `dataSendPoint.returnValueProvision`

Imposition time: CP: `IT_CpgExe`

⌈All `RunnableEntity.dataSendPoint` that refer to the same `accessedVariable` shall define the identical value for attribute `returnValueProvision`.⌋

[constr_1774] Value of attribute `dataReceivePointByArgument.returnValueProvision`

Imposition time: CP: `IT_CpgExe`

⌈All `RunnableEntity.dataReceivePointByArgument` that refer to the same `accessedVariable` shall define the identical value for attribute `returnValueProvision`.⌋

[constr_1775] Value of attribute `serverCallPoint.returnValueProvision`

Imposition time: CP: `IT_CpgExe`

⌈All `RunnableEntity.serverCallPoint` that refer to the same `operation` shall define the identical value of attribute `returnValueProvision`.⌋

[constr_1776] Value of attribute `asynchronousServerCallResultPoint.returnValueProvision`

Imposition time: CP: `IT_CpgExe`

⌈All `RunnableEntity.asynchronousServerCallResultPoint` that refer to the same `AsynchronousServerCallPoint.operation` shall define the identical value of attribute `returnValueProvision`.⌋

[constr_1777] Value of attribute `externalTriggeringPoint.returnValueProvision`

Imposition time: CP: `IT_CpgExe`

⌈All `RunnableEntity.externalTriggeringPoint` that refer to the same `trigger` shall define the identical value of attribute `returnValueProvision`.⌋

[constr_1778] Value of attribute modeSwitchPoint.returnValueProvision

Imposition time: CP: IT_CpgExe

⌈All RunnableEntity.modeSwitchPoint that refer to the same modeGroup shall define the identical value of attribute returnValueProvision.⌋

[constr_1779] Scope of the definition of an AbstractRuleBasedValueSpecification

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈An AbstractRuleBasedValueSpecification shall only be defined in the context of an ArrayValueSpecification or a ConstantSpecification. If the AbstractRuleBasedValueSpecification is defined in the context of a ConstantSpecification then a reference to this ConstantSpecification shall only be created in the context of an ArrayValueSpecification.⌋

[constr_1783] Existence of primitive attribute ImplementationDataTypeElement.arrayImplPolicy

Imposition time: CP: IT_CpgExe

⌈Attribute ImplementationDataTypeElement.arrayImplPolicy shall only exist if the enclosing ImplementationDataType or ImplementationDataTypeElement is of category ARRAY.⌋

[constr_1860] Multiplicity of DelegationSwConnector.innerPort

Imposition time: CP: IT_CompSwcT

⌈For each DelegationSwConnector, the reference DelegationSwConnector.innerPort shall exist.⌋

[constr_1861] Multiplicity of DelegationSwConnector.outerPort

Imposition time: CP: IT_CompSwcT

⌈For each DelegationSwConnector, the reference DelegationSwConnector.outerPort shall exist.⌋

[constr_1862] Multiplicity of PassThroughSwConnector.requiredOuterPort

Imposition time: CP: IT_CompSwcT, AP: IT_BefAraApiGen

⌈For each PassThroughSwConnector, the reference PassThroughSwConnector.requiredOuterPort shall exist.⌋

[constr_1863] Multiplicity of PassThroughSwConnector.providedOuterPort

Imposition time: CP: IT_CompSwcT, AP: IT_BefAraApiGen

⌈For each PassThroughSwConnector, the reference PassThroughSwConnector.providedOuterPort shall exist.⌋

[constr_1864] Multiplicity of `InstantiationRTEEventProps.refinedEvent`

Imposition time: CP: `IT_RteGen`

⌈For each `InstantiationRTEEventProps`, the instance-reference in the role `refinedEvent` shall exist.⌋

[constr_1865] Existence of `InvalidationPolicy.dataElement`

Imposition time: CP: `IT_CpgExe`

⌈For each `InvalidationPolicy`, the reference `InvalidationPolicy.dataElement` shall exist.⌋

[constr_1866] Existence of `MetaDataItem.length`

Imposition time: CP: `IT_CpgExe`

⌈For each `MetaDataItem`, attribute `length` shall exist.⌋

[constr_1867] Existence of `MetaDataItem.metaDataItemType`

Imposition time: CP: `IT_CpgExe`

⌈For each `MetaDataItem`, attribute `metaDataItemType` shall exist.⌋

[constr_1868] Existence of `MetaDataItemSet.dataElement`

Imposition time: CP: `IT_CpgExe`

⌈For each `MetaDataItemSet` that aggregates at least one `metaDataItem`, at least one reference to a `dataElement` shall exist.⌋

[constr_1869] Existence of attribute `ArgumentDataPrototype.direction`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈For each `ArgumentDataPrototype`, attribute `direction` shall be defined.⌋

[constr_1871] Existence of attribute `ModeRequestTypeMap.implementationDataType`

Imposition time: CP: `IT_CpgExe`

⌈For each `ModeRequestTypeMap`, attribute `implementationDataType` shall exist.⌋

[constr_1872] Existence of attribute `ModeRequestTypeMap.modeGroup`

Imposition time: CP: `IT_CpgExe`

⌈For each `ModeRequestTypeMap`, attribute `modeGroup` shall exist.⌋

[constr_1873] Existence of `DataPrototypeMapping.firstDataPrototype`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈For each `DataPrototypeMapping`, the reference in the role `firstDataPrototype` shall exist.⌋

[constr_1874] Existence of DataPrototypeMapping.secondDataPrototype

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each DataPrototypeMapping, the reference in the role secondDataPrototype shall exist.⌋

[constr_1875] Existence of reference in the role ClientServerOperationMapping.firstOperation

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each ClientServerOperationMapping, the reference in the role firstOperation shall exist.⌋

[constr_1876] Existence of reference in the role ClientServerOperationMapping.secondOperation

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each ClientServerOperationMapping, the reference in the role secondOperation shall exist.⌋

[constr_1877] Existence of reference ModeDeclarationGroupPrototypeMapping.firstModeGroup

Imposition time: CP: IT_RteGen

⌈For each ModeDeclarationGroupPrototypeMapping, the reference in the role firstModeGroup shall exist.⌋

[constr_1878] Existence of reference ModeDeclarationGroupPrototypeMapping.secondModeGroup

Imposition time: CP: IT_RteGen

⌈For each ModeDeclarationGroupPrototypeMapping, the reference in the role secondModeGroup shall exist.⌋

[constr_1879] Existence of reference ModeDeclarationMapping.firstMode

Imposition time: CP: IT_RteGen

⌈For each ModeDeclarationMapping, at least one reference firstMode shall exist.⌋

[constr_1880] Existence of reference ModeDeclarationMapping.secondMode

Imposition time: CP: IT_RteGen

⌈For each ModeDeclarationMapping, the reference secondMode shall exist.⌋

[constr_1881] Existence of reference TriggerMapping.firstTrigger

Imposition time: CP: IT_RteGen

⌈For each TriggerMapping, the reference firstTrigger shall exist.⌋

[constr_1882] Existence of reference TriggerMapping.secondTrigger

Imposition time: CP: IT_RteGen

⌈For each TriggerMapping, the reference secondTrigger shall exist.⌋

[constr_1883] Existence of ApplicationCompositeDataTypeSubElementRef.applicationCompositeElement

Imposition time: CP: IT_RteGen

⌈For each ApplicationCompositeDataTypeSubElementRef, the reference applicationCompositeElement shall exist.⌋

[constr_1884] Existence of attribute TextTableMapping.identicalMapping

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each TextTableMapping, the attribute identicalMapping shall exist.⌋

[constr_1885] Existence of attribute TextTableMapping.mappingDirection

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each TextTableMapping, the attribute mappingDirection shall exist.⌋

[constr_1886] Existence of attribute TextTableValuePair.firstValue

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each TextTableValuePair, the attribute firstValue shall exist.⌋

[constr_1887] Existence of attribute TextTableValuePair.secondValue

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈For each TextTableValuePair, the attribute secondValue shall exist.⌋

[constr_1888] Existence of attribute DataTransformation.executeDespiteDataUnavailability

Imposition time: CP: IT_RteGen

⌈For each DataTransformation, the attribute executeDespiteDataUnavailability shall exist.⌋

[constr_1889] Existence of attribute QueuedReceiverComSpec.queueLength

Imposition time: CP: IT_CpgExe

⌈For each QueuedReceiverComSpec, attribute queueLength shall exist.⌋

[constr_1890] Existence of attribute DataFilter.dataFilterType

Imposition time: CP: IT_CpgExe

⌈For each DataFilter, attribute dataFilterType shall exist.⌋

[constr_1892] Existence of attribute `TransmissionAcknowledgementRequest.timeout`

Imposition time: CP: `IT_CpgExe`

⌈For each `TransmissionAcknowledgementRequest`, attribute `timeout` shall exist.⌋

[constr_1894] Existence of attribute `ModeSwitchSenderComSpec.queueLength`

Imposition time: CP: `IT_CpgExe`

⌈For each `ModeSwitchSenderComSpec`, attribute `queueLength` shall exist.⌋

[constr_1895] Existence of attribute `ModeSwitchSenderComSpec.modeGroup`

Imposition time: CP: `IT_CpgExe`

⌈For each `ModeSwitchSenderComSpec`, attribute `modeGroup` shall exist.⌋

[constr_1896] Existence of attribute `ModeSwitchReceiverComSpec.modeGroup`

Imposition time: CP: `IT_CpgExe`

⌈For each `ModeSwitchReceiverComSpec`, attribute `modeGroup` shall exist.⌋

[constr_1897] Existence of reference `ParameterProvideComSpec.parameter`

Imposition time: CP: `IT_CpgExe`

⌈For each `ParameterProvideComSpec`, the reference `parameter` shall exist.⌋

[constr_1898] Existence of reference `ParameterRequireComSpec.parameter`

Imposition time: CP: `IT_CpgExe`

⌈For each `ParameterRequireComSpec`, the reference `parameter` shall exist.⌋

[constr_1899] Existence of reference `NvRequireComSpec.variable`

Imposition time: CP: `IT_CpgExe`

⌈For each `NvRequireComSpec`, the reference `variable` shall exist.⌋

[constr_1900] Existence of reference `NvProvideComSpec.variable`

Imposition time: CP: `IT_CpgExe`

⌈For each `NvProvideComSpec`, the reference `variable` shall exist.⌋

[constr_1903] Existence of reference `DataTypeMap.applicationDataType`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `DataTypeMap`, reference `applicationDataType` shall exist.⌋

[constr_1904] Existence of reference `DataTypeMap.implementationDataType`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `DataTypeMap`, reference `implementationDataType` shall exist.⌋

[constr_1905] Existence of attribute SwTextProps.arraySizeSemantics

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each SwTextProps, attribute arraySizeSemantics shall exist.⌋

[constr_1906] Existence of attribute SwTextProps.swMaxTextSize

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each SwTextProps, attribute swMaxTextSize shall exist.⌋

[constr_1907] Existence of attribute ApplicationArrayDataType.element

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each ApplicationArrayDataType, the aggregation of ApplicationArrayElement in the role element shall exist.⌋

[constr_1908] Existence of attribute ApplicationRecordDataType.element

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each ApplicationRecordDataType, the aggregation of ApplicationRecordElement in the role element shall exist.⌋

[constr_1909] Existence of attribute ImplementationProps.symbol

Imposition time: CP: IT_CpgExe

⌈For each ImplementationProps, the attribute symbol shall exist.⌋

[constr_1910] Existence of attribute BaseType.baseTypeDefinition

Imposition time: CP: IT_CpgExe

⌈For each BaseType (which will be utilized in the form of SwBaseType), the aggregation in the role baseTypeDefinition shall exist.⌋

[constr_1911] Existence of ArVariableInImplementationDataInstanceRef.targetDataPrototype

Imposition time: CP: IT_CpgExe

⌈For each ArVariableInImplementationDataInstanceRef, the reference targetDataPrototype shall exist.⌋

[constr_1912] Existence of reference ArParameterInImplementationDataInstanceRef.targetDataPrototype

Imposition time: CP: IT_CpgExe

⌈For each ArParameterInImplementationDataInstanceRef, the reference targetDataPrototype shall exist.⌋

[constr_1913] Existence of attribute CompuRationalCoeffs.compuDenominator

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each CompuRationalCoeffs, the attribute compuDenominator shall exist.⌋

[constr_1914] Existence of attribute `CompuRationalCoeffs.compuNumerator`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `CompuRationalCoeffs`, the attribute `compuNumerator` shall exist.⌋

[constr_1915] Existence of attribute `PhysicalDimensionMapping.firstPhysicalDimension`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `PhysicalDimensionMapping`, attribute `firstPhysicalDimension` shall exist.⌋

[constr_1916] Existence of attribute `PhysicalDimensionMapping.secondPhysicalDimension`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `PhysicalDimensionMapping`, attribute `secondPhysicalDimension` shall exist.⌋

[constr_1917] Existence of `ConstantSpecification.valueSpec`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `ConstantSpecification`, the aggregation of `ValueSpecification` in the role `valueSpec` shall exist.⌋

[constr_1918] Existence of `RecordValueSpecification.field`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `RecordValueSpecification`, the aggregation of meta-class `ValueSpecification` in the role `field` shall exist.⌋

[constr_1919] Existence of `TextValueSpecification.value`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `TextValueSpecification`, attribute `value` shall exist.⌋

[constr_1920] Existence of `NumericalValueSpecification.value`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `NumericalValueSpecification`, attribute `value` shall exist.⌋

[constr_1921] Existence of `ReferenceValueSpecification.referenceValue`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `ReferenceValueSpecification`, attribute `referenceValue` shall exist.⌋

[constr_1922] Existence of ApplicationRuleBasedValueSpecification.category

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each ApplicationRuleBasedValueSpecification, attribute category shall exist.⌋

[constr_1923] Existence of RuleBasedAxisCont.ruleBasedValues

Imposition time: CP: IT_CpgExe

⌈For each RuleBasedAxisCont, attribute ruleBasedValues shall exist.⌋

[constr_1924] Existence of RuleBasedValueCont.ruleBasedValues

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each RuleBasedValueCont, attribute ruleBasedValues shall exist.⌋

[constr_1925] Existence of NumericalRuleBasedValueSpecification.ruleBasedValues

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each NumericalRuleBasedValueSpecification, attribute ruleBasedValues shall exist.⌋

[constr_1926] Existence of RuleBasedValueSpecification.rule

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each RuleBasedValueSpecification, attribute rule shall exist.⌋

[constr_1927] Existence of RuleBasedValueSpecification.arguments

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each RuleBasedValueSpecification, the aggregation of RuleArguments in the role arguments shall exist.⌋

[constr_1928] Existence of CompositeRuleBasedValueSpecification.rule

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each CompositeRuleBasedValueSpecification, attribute rule shall exist.⌋

[constr_1929] Existence of CompositeRuleBasedValueSpecification.argument

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each CompositeRuleBasedValueSpecification, the aggregation of CompositeValueSpecification in the role argument shall exist.⌋

[constr_1930] Existence of ConstantReference.constant

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each ConstantReference, attribute constant shall exist.⌋

[constr_1931] Existence of ConstantSpecificationMapping.applConstant

Imposition time: CP: IT_CpgExe

⌈For each ConstantSpecificationMapping, the reference to meta-class ConstantSpecification in the role applConstant shall exist.⌋

[constr_1932] Existence of ConstantSpecificationMapping.implConstant

Imposition time: CP: IT_CpgExe

⌈For each ConstantSpecificationMapping, the reference to meta-class ConstantSpecification in the role implConstant shall exist.⌋

[constr_1933] Existence of the role CalibrationParameterValue.initializedParameter

Imposition time: CP: IT_CpgExe

⌈For each CalibrationParameterValue, the reference to meta-class ConstantSpecification in the role initializedParameter shall exist.⌋

[constr_1935] Existence of attribute SwcInternalBehavior.supportsMultipleInstantiation

Imposition time: CP: IT_CpgExe

⌈For each SwcInternalBehavior, attribute supportsMultipleInstantiation shall exist.⌋

[constr_1936] Existence of attribute RunnableEntity.symbol

Imposition time: CP: IT_CpgExe

⌈For each RunnableEntity, attribute symbol shall exist.⌋

[constr_1938] Existence of attribute RunnableEntityArgument.symbol

Imposition time: CP: IT_CpgExe

⌈For each RunnableEntityArgument, attribute symbol shall exist.⌋

[constr_1939] Existence of attribute ExecutableEntityActivationReason.bitPosition

Imposition time: CP: IT_CpgExe

⌈For each ExecutableEntityActivationReason, attribute bitPosition shall exist.⌋

[constr_1940] Existence of attribute AsynchronousServerCallReturnsEvent.eventSource

Imposition time: CP: IT_CpgExe

⌈For each AsynchronousServerCallReturnsEvent, attribute eventSource shall exist.⌋

[constr_1941] Existence of attribute DataSendCompletedEvent.eventSource

Imposition time: CP: IT_CpgExe

⌈For each DataSendCompletedEvent, attribute eventSource shall exist.⌋

[constr_1942] Existence of attribute DataWriteCompletedEvent.eventSource

Imposition time: CP: IT_CpgExe

⌈For each DataWriteCompletedEvent, attribute eventSource shall exist.⌋

[constr_1943] Existence of attribute DataReceivedEvent.data

Imposition time: CP: IT_CpgExe

⌈For each DataReceivedEvent, attribute data shall exist.⌋

[constr_1944] Existence of attribute DataReceiveErrorEvent.data

Imposition time: CP: IT_CpgExe

⌈For each DataReceiveErrorEvent, attribute data shall exist.⌋

[constr_1945] Existence of attribute OperationInvokedEvent.operation

Imposition time: CP: IT_CpgExe

⌈For each OperationInvokedEvent, attribute operation shall exist.⌋

[constr_1946] Existence of attribute SwcModeSwitchEvent.activation

Imposition time: CP: IT_RteGen

⌈For each SwcModeSwitchEvent, attribute activation shall exist.⌋

[constr_1947] Existence of reference SwcModeSwitchEvent.mode

Imposition time: CP: IT_RteGen

⌈For each SwcModeSwitchEvent, the reference to ModeDeclaration in the role mode shall exist.⌋

[constr_1948] Existence of attribute ModeSwitchedAckEvent.eventSource

Imposition time: CP: IT_RteGen

⌈For each ModeSwitchedAckEvent, attribute eventSource shall exist.⌋

[constr_1949] Existence of attribute ExternalTriggerOccurredEvent.trigger

Imposition time: CP: IT_RteGen

⌈For each ExternalTriggerOccurredEvent, attribute trigger shall exist.⌋

[constr_1950] Existence of attribute `InternalTriggerOccurredEvent.eventSource`

Imposition time: CP: `IT_RteGen`

⌈For each `InternalTriggerOccurredEvent`, the attribute `eventSource` shall exist.⌋

[constr_1951] Existence of attribute `WaitPoint.timeout`

Imposition time: CP: `IT_RteGen`

⌈For each `WaitPoint`, attribute `timeout` shall exist.⌋

[constr_1952] Existence of reference `WaitPoint.trigger`

Imposition time: CP: `IT_CpgExe`

⌈For each `WaitPoint`, the reference to `RTEEvent` in the role `trigger` shall exist.⌋

[constr_1953] Existence of attribute `SwcExclusiveAreaPolicy.apiPrinciple`

Imposition time: CP: `IT_RteGen`

⌈For each `SwcExclusiveAreaPolicy` that refers to an `exclusiveArea`, attribute `apiPrinciple` shall exist.⌋

[constr_1954] Existence of attribute `VariableAccess.accessedVariable`

Imposition time: CP: `IT_CpgExe`

⌈For each `VariableAccess`, attribute `accessedVariable` shall exist.⌋

[constr_1955] Existence of attribute `ServerCallPoint.operation`

Imposition time: CP: `IT_CpgExe`

⌈For each `ServerCallPoint`, attribute `operation` shall exist.⌋

[constr_1956] Existence of attribute `ServerCallPoint.timeout`

Imposition time: CP: `IT_RteGen`

⌈For each `ServerCallPoint`, attribute `timeout` shall exist.⌋

[constr_1957] Existence of reference `AsynchronousServerCallResultPoint.asynchronousServerCallPoint`

Imposition time: CP: `IT_CpgExe`

⌈For each `AsynchronousServerCallResultPoint`, the reference to `AsynchronousServerCallPoint` in the role `asynchronousServerCallPoint` shall exist.⌋

[constr_1958] Existence of attribute `ParameterAccess.accessedParameter`

Imposition time: CP: `IT_CpgExe`

⌈For each `ParameterAccess`, attribute `accessedParameter` shall exist.⌋

[constr_1959] Existence of attribute `InstantiationDataDefProps.swDataDefProps`

Imposition time: CP: `IT_CpgExe`

⌈For each `InstantiationDataDefProps`, attribute `swDataDefProps` shall exist.⌋

[constr_1960] Existence of attribute `PortAPIOption.port`

Imposition time: CP: `IT_CpgExe`

⌈For each `PortAPIOption`, attribute `port` shall exist.⌋

[constr_1961] Existence of attribute `PortDefinedArgumentValue.value`

Imposition time: CP: `IT_RteGen`

⌈For each `PortDefinedArgumentValue`, attribute `value` shall exist.⌋

[constr_1962] Existence of attribute `PortDefinedArgumentValue.valueType`

Imposition time: CP: `IT_RteGen`

⌈For each `PortDefinedArgumentValue`, attribute `valueType` shall exist.⌋

[constr_1963] Existence of attribute `CommunicationBufferLocking.supportBufferLocking`

Imposition time: CP: `IT_RteGen`

⌈For each `CommunicationBufferLocking`, attribute `supportBufferLocking` shall exist.⌋

[constr_1964] Existence of attribute `PerInstanceMemory.type`

Imposition time: CP: `IT_CpgExe`

⌈For each `PerInstanceMemory`, attribute `type` shall exist.⌋

[constr_1965] Existence of attribute `PerInstanceMemory.typeDefinition`

Imposition time: CP: `IT_CpgExe`

⌈For each `PerInstanceMemory`, attribute `typeDefinition` shall exist.⌋

[constr_1966] Existence of attribute `Implementation.swVersion`

Imposition time: CP: `IT_RteGen`

⌈For each `Implementation`, attribute `swVersion` shall exist.⌋

[constr_1967] Existence of attribute `Implementation.vendorId`

Imposition time: CP: `IT_RteGen`

⌈For each `Implementation`, attribute `vendorId` shall exist.⌋

[constr_1968] Existence of attribute Implementation.codeDescriptor

Imposition time: CP: IT_RteGen

⌈For each Implementation, at least one aggregation of Code in the role codeDescriptor shall exist.⌋

[constr_1969] Existence of attribute SwcImplementation.behavior

Imposition time: CP: IT_RteGen

⌈For each SwcImplementation, attribute behavior shall exist.⌋

[constr_1970] Existence of attribute PerInstanceMemorySize.alignment

Imposition time: CP: IT_RteGen

⌈For each PerInstanceMemorySize, attribute alignment shall exist.⌋

[constr_1971] Existence of attribute PerInstanceMemorySize.perInstanceMemory

Imposition time: CP: IT_RteGen

⌈For each PerInstanceMemorySize, the reference to PerInstanceMemory in the role perInstanceMemory shall exist.⌋

[constr_1972] Existence of attribute PerInstanceMemorySize.size

Imposition time: CP: IT_RteGen

⌈For each PerInstanceMemorySize, attribute size shall exist.⌋

[constr_1973] Existence of attribute ModeDeclarationGroup.initialMode

Imposition time: CP: IT_CpgExe

⌈For each ModeDeclarationGroup, the reference to ModeDeclaration in the role initialMode shall exist.⌋

[constr_1974] Existence of attribute ModeDeclarationGroup.modeDeclaration

Imposition time: CP: IT_CpgExe

⌈For each ModeDeclarationGroup, at least one ModeDeclaration shall be aggregated in the role modeDeclaration.⌋

[constr_1975] Existence of attribute ModeTransition.enteredMode

Imposition time: CP: IT_RteGen

⌈For each ModeTransition, the reference to ModeDeclaration in the role enteredMode shall exist.⌋

[constr_1976] Existence of attribute ModeTransition.exitedMode

Imposition time: CP: IT_RteGen

⌈For each ModeTransition, the reference to ModeDeclaration in the role exitedMode shall exist.⌋

[constr_1977] Existence of attribute ModeErrorBehavior.errorReaction-Policy

Imposition time: CP: IT_RteGen

⌈For each ModeErrorBehavior, the attribute errorReactionPolicy shall exist.⌋

[constr_1978] Existence of attribute SwcModeManagerErrorEvent.modeGroup

Imposition time: CP: IT_RteGen

⌈For each SwcModeManagerErrorEvent, the instance reference to Mode-Declaration in the role modeGroup shall exist.⌋

[constr_1979] Existence of the reference SwcBswMapping.bswBehavior

Imposition time: CP: IT_RteGen

⌈For each SwcBswMapping, the reference to BswInternalBehavior in the role bswBehavior shall exist.⌋

[constr_1980] Existence of the reference SwcBswMapping.swcBehavior

Imposition time: CP: IT_RteGen

⌈For each SwcBswMapping, the reference to BswInternalBehavior in the role swcBehavior shall exist.⌋

[constr_1981] Existence of attribute NvBlockDescriptor.nvBlockNeeds

Imposition time: CP: IT_RteGen

⌈For each NvBlockDescriptor, attribute nvBlockNeeds shall exist.⌋

[constr_1982] Existence of attribute ModeSwitchEventTriggeredActivity.role

Imposition time: CP: IT_RteGen

⌈For each ModeSwitchEventTriggeredActivity, attribute role shall exist.⌋

[constr_1983] Existence of attribute ModeSwitchEventTriggeredActivity.swcModeSwitchEvent

Imposition time: CP: IT_RteGen

⌈For each ModeSwitchEventTriggeredActivity, attribute swcModeSwitch-Event shall exist.⌋

[constr_1984] Existence of instance reference NvBlockDataMapping.nvRam-BlockElement

Imposition time: CP: IT_RteGen

⌈For each NvBlockDataMapping, the instance reference to ModeDeclaration in the role nvRamBlockElement shall exist.⌋

[constr_1985] Existence of the reference in the role SupervisedEntityNeeds.toleratedFailedCycles

Imposition time: CP: IT_RteGen

⌈For each SupervisedEntityNeeds, the reference to BswInternalBehavior in the role toleratedFailedCycles shall exist.⌋

[constr_1986] Existence of the reference DiagnosticRoutineNeeds.diagRoutineType

Imposition time: CP: IT_RteGen

⌈For each DiagnosticRoutineNeeds, the attribute diagRoutineType shall exist.⌋

[constr_1987] Existence of instance reference RapidPrototypingScenario.hostSystem

Imposition time: CP: IT_RteGen

⌈For each RapidPrototypingScenario, the instance reference to ModeDeclaration in the role hostSystem shall exist.⌋

[constr_2000] Compatibility of ClientServerOperations triggering the same RunnableEntity

Imposition time: CP: IT_CpgExe

⌈The ClientServerOperations are considered compatible if

- the number of arguments (which can be ArgumentDataPrototypes or related PortDefinedArgumentValues) is equal and
- the corresponding arguments (i.e. first argument on both sides, second argument on both sides, etc.) are compatible or both are typed by "new-world" Variable-Size Array Data Types where the data types of the array elements are compatible (but the array sizes may differ).
- and the respective values of PortAPIOption.errorHandling are identical.

In particular, this means that:

- for combinations of ArgumentDataPrototypes and ArgumentDataPrototypes where the serverArgumentImplPolicy is set to useArgumentType the referred ImplementationDataTypes shall be compatible.

  In the case of data types of category STRUCTURE all by-order matching ImplementationDataTypeElements shall be named equally.
- for combinations of PortDefinedArgumentValues and ArgumentDataPrototypes where the serverArgumentImplPolicy is set to useArgumentType the referred ImplementationDataTypes shall be compatible.

In the case of ImplementationDataTypeElements of category STRUCTURE all by-order matching ImplementationDataTypeElements of the structure shall be named equally.

- for ArgumentDataPrototypes where the serverArgumentImplPolicy is set to useVoid an arbitrary ImplementationDataType is referred to.

In addition, it is required that the **return value defined on both sides shall match** (in terms of Std_ReturnType vs. void) and also the possibleErrors are compatible.⌋

[constr_2002] Referenced VariableDataPrototype from AutosarVariableRef of VariableAccess in role dataReadAccess

Imposition time: CP: IT_CpgExe

⌈A VariableAccess in the role dataReadAccess shall refer to an RPortPrototype or PRPortPrototype that is typed by either a SenderReceiverInterface or a NvDataInterface.⌋

[constr_2003] Referenced VariableDataPrototype from AutosarVariableRef of VariableAccess in role dataWriteAccess

Imposition time: CP: IT_CpgExe

⌈A VariableAccess in the role dataWriteAccess shall refer to a PPortPrototype or PRPortPrototype that is typed by either a SenderReceiverInterface or a NvDataInterface.⌋

[constr_2004] Referenced VariableDataPrototype from AutosarVariableRef of VariableAccess in role dataSendPoint

Imposition time: CP: IT_CpgExe

⌈A VariableAccess in the role dataSendPoint shall refer to a PPortPrototype or PRPortPrototype that is typed by either a SenderReceiverInterface or a NvDataInterface.⌋

[constr_2005] Referenced VariableDataPrototype from AutosarVariableRef of VariableAccess in role dataReceivePointByValue or dataReceivePointByArgument

Imposition time: CP: IT_CpgExe

⌈A VariableAccess in the role dataReceivePointByValue or dataReceivePointByArgument shall refer to an RPortPrototype or PRPortPrototype that is typed by either a SenderReceiverInterface or an NvDataInterface.⌋

[constr_2006] Number of AsynchronousServerCallResultPoint referencing to one AsynchronousServerCallPoint

Imposition time: CP: IT_CpgExe

⌈The AsynchronousServerCallPoint may be referenced by at most one AsynchronousServerCallResultPoint.

If the reference exists, this means that only the RunnableEntity with this AsynchronousServerCallResultPoint can fetch the result of the asynchronous server invocation of this particular AsynchronousServerCallPoint.⌋

[constr_2007] Consistency of typeDefinition attribute

Imposition time: CP: IT_CpgExe

⌈All PerInstanceMemorys of the same SwcInternalBehavior with identical type attribute shall define an identical typeDefinition attribute as well.⌋

[constr_2009] Supported kinds of PortPrototypes of a NvBlockSwComponentType

Imposition time: CP: IT_RteGen

⌈With respect to external communication, NvBlockSwComponentType is limited to the definition of the following kinds of PortPrototype:

- PortPrototypes typed by either NvDataInterfaces or ClientServerInterfaces
- RPortPrototypes typed by ModeSwitchInterfaces

⌋

[constr_2010] Connections between SwComponentPrototypes of type NvBlockSwComponentType

Imposition time: CP: IT_RteGen

⌈The existence of SwConnectors that refer to PortPrototypes belonging to SwComponentPrototypes where both are typed by NvBlockSwComponentType is not permitted.⌋

[constr_2011] Connections between SwComponentPrototypes typed by NvBlockSwComponentType and SwComponentPrototypes typed by other AtomicSwComponentTypes

Imposition time: CP: IT_RteGen

⌈A PortPrototype typed by an NvDataInterface owned by a SwComponentPrototype typed by an NvBlockSwComponentType shall be connected to a PortPrototype typed by **either** an NvDataInterface **or** a SenderReceiverInterface owned by a SwComponentPrototype that is typed by another subclass of AtomicSwComponentType.⌋

[constr_2012] Compatibility of ImplementationDataTypes used for ramBlock and romBlock

Imposition time: CP: IT_RteGen

⌈The ramBlock and the romBlock shall have compatible ImplementationDataTypes to ensure that the NVRAM Block default values in the ROM Block can be copied into the RAM Block.⌋

[constr_2013] Compatibility of ImplementationDataTypes for NvBlockDataMapping

Imposition time: CP: IT_RteGen

⌈Unless both the attribute bitfieldTextTableMaskNvBlockDescriptor and attribute bitfieldTextTableMaskPortPrototype is defined in the context of a given NvBlockDataMapping, the NvBlockDataMapping is only valid if the ImplementationDataType of the referenced VariableDataPrototype or ImplementationDataTypeElement in the role nvRamBlockElement is compatible to the ImplementationDataType used to type the DataPrototype aggregated by NvBlockDataMapping in the role writtenNvData, writtenReadNvData, or readNvData.⌋

[constr_2014] Limitation of NvBlockDescriptor.clientServerPort.role

Imposition time: CP: IT_RteGen

⌈The value of attribute NvBlockDescriptor.clientServerPort.role shall be set to a valid name of one of the standardized AUTOSAR (client/server) Interfaces used for the NVRAM Manager, as described by [TPS_SWCT_02501], [TPS_SWCT_02502], [TPS_SWCT_02503] and [TPS_SWCT_02504].⌋

[constr_2015] Limitation of SwcInternalBehavior of a NvBlockSwComponentType

Imposition time: CP: IT_RteGen

⌈The SwcInternalBehavior of a NvBlockSwComponentType is only permitted to define

- OperationInvokedEvents
- RunnableEntitys triggered by OperationInvokedEvents (server RunnableEntitys)
- RunnableEntitys which defines only the mandatory attributes symbol and canBeInvokedConcurrently
- PortAPIOptions defining PortDefinedArgumentValues
- TimingEvents (which may include references to ModeDeclarations in the role disabledMode)
- DataReceivedEvents (which may include references to ModeDeclarations in the role disabledMode)
- SwcModeSwitchEvents
- RunnableEntitys triggered by TimingEvents
- RunnableEntitys triggered by DataReceivedEvents
- RunnableEntitys triggered by SwcModeSwitchEvents
- DataTypeMappingSet

⌋

[constr_2016] Connections between SwComponentPrototypes of type ServiceProxySwComponentType

Imposition time: CP: IT_RteGen

⌈A connection between PortPrototypes belonging to SwComponentPrototypes where both are typed by ServiceProxySwComponentType is not permitted.⌋

[constr_2017] Ports of ServiceProxySwComponentTypes

Imposition time: CP: IT_RteGen

⌈ServiceProxySwComponentType is only permitted to define

- RPortPrototypes that are typed by SenderReceiverInterface or
- PortPrototypes that are typed by a PortInterface where the isService attribute is set to true.

⌋

[constr_2018] Supported remote communication of a ServiceProxySwComponentType

Imposition time: CP: IT_RteGen

⌈For remote communication, ServiceProxySwComponentType can have only RPortPrototypes typed by SenderReceiverInterfaces in a 1:n communication scenario.⌋

[constr_2019] ServiceSwComponentType shall have service ports only

Imposition time: CP: IT_RteGen

⌈In the case of ServiceSwComponentType, all aggregated PortPrototypes need to have an ≪isOfType≫ relationship to a PortInterface which has its isService attribute set to true.

The exceptions described in

- [TPS_SWCT_01572],
- [TPS_SWCT_01579],
- [TPS_SWCT_01831] and
- [TPS_SWCT_01580]

apply.⌋

[constr_2020] dataReadAccess cannot be used for queued communication

Imposition time: CP: IT_CpgExe

⌈The swImplPolicy of the VariableDataPrototype referenced by a VariableAccess in role dataReadAccess shall **not** be set to queued.⌋

[constr_2021] `WaitPoint` referencing a `DataReceivedEvent` cannot be used for non-queued communication

Imposition time: CP: `IT_CpgExe`

⌈A `WaitPoint` referencing a `DataReceivedEvent` is permitted **if and only if** the `swImplPolicy` of the `VariableDataPrototype` referenced by this `DataReceivedEvent` is set to `queued`.⌋

[constr_2022] Mutually exclusive use of `SynchronousServerCallPoint`s and `AsynchronousServerCallPoint`s

Imposition time: CP: `IT_CpgExe`

⌈A `ClientServerOperation` of a particular `RPortPrototype` shall be referenced by either a `SynchronousServerCallPoint` or an `AsynchronousServerCallPoint`, but not by both.⌋

[constr_2023] Consistency of `timeout` values

Imposition time: CP: `IT_RteGen`

⌈The `timeout` values of all `ServerCallPoint`s referencing the same instance of `ClientServerOperation` in a `RPortPrototype` shall be identical.⌋

[constr_2024] `enableTakeAddress` is restricted to single instantiation

Imposition time: CP: `IT_CpgExe`

⌈The definition of a `PortAPIOption` with `enableTakeAddress` set to `true` is only permitted for software-components where the attribute `SwcInternalBehavior.supportsMultipleInstantiation` is set to `false`.⌋

[constr_2026] Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `writtenLocalVariable` and `readLocalVariable`

Imposition time: CP: `IT_CpgExe`

⌈A `VariableDataPrototype` in the `localVariable` reference shall be owned by the same `SwcInternalBehavior` as this `RunnableEntity` belongs to, and the referenced `VariableDataPrototype` shall be defined in the role `implicitInterRunnableVariable` or `explicitInterRunnableVariable`.⌋

[constr_2027] `SwcServiceDependency` shall be defined for service ports only

Imposition time: CP: `IT_RteGen`

⌈A `PortPrototype` that is referenced by a `SwcServiceDependency` via `assignedPort` or via `assignedData` shall be typed by a `PortInterface` that has `isService` set to `true`.

This rule does **not** apply to `PortPrototype`s referenced by a `RoleBasedPortAssignment` where the attribute `role` is set to any of the following values:

- `NvMService`

- NvMNotifyJobFinished
- NvMNotifyInitBlock
- NvMAdmin
- NvMMirror
- NvDataPort

Furthermore, the rule does **not** apply to the case described in [TPS_SWCT_01579], [TPS_SWCT_01831], [TPS_SWCT_01580], and [TPS_SWCT_01572].⌋

[constr_2028] staticMemory is restricted to single instantiation

Imposition time: CP: IT_RteGen

⌈The staticMemory is only supported if the attribute supportsMultiple-Instantiation of the owning SwcInternalBehavior is set to false.⌋

[constr_2030] AsynchronousServerCallResultPoint combined with WaitPoint shall belong to the same RunnableEntity

Imposition time: CP: IT_CpgExe

⌈A WaitPoint referencing a AsynchronousServerCallReturnsEvent as well as a AsynchronousServerCallResultPoint referenced by said Asynchronous-ServerCallReturnsEvent shall be aggregated by the same RunnableEntity.⌋

[constr_2031] Value of TimingEvent.period shall be greater than 0

Imposition time: CP: IT_RteGen

⌈Attribute TimingEvent.period shall exist and its value shall be greater than 0.⌋

[constr_2033] Timeout of DataSendCompletedEvent

Imposition time: CP: IT_RteGen

⌈The timeout value of a WaitPoint associated with a DataSendCompleted-Event shall have the same value as the corresponding value of Transmission-AcknowledgementRequest.timeout.⌋

[constr_2034] SwAddrMethod referenced by RunnableEntitys, BswCalled-Entitys, or BswSchedulableEntitys

Imposition time: CP: IT_CpgExe

⌈RunnableEntitys, BswCalledEntitys, and BswSchedulableEntitys shall not reference a SwAddrMethod which attribute memoryAllocationKeyword-Policy is set to addrMethodShortNameAndAlignment.⌋

[constr_2035] swImplPolicy for VariableDataPrototype in SenderReceiverInterface

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype owned by a SenderReceiverInterface shall be either standard, queued, or measurementPoint.⌋

[constr_2036] swImplPolicy for VariableDataPrototype in NvDataInterface

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype owned by a NvDataInterface shall be standard.⌋

[constr_2037] swImplPolicy for VariableDataPrototype in the role ramBlock

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype aggregated in the role NvBlockDescriptor.ramBlock shall be standard.⌋

[constr_2038] swImplPolicy for VariableDataPrototype in the role implicitInterRunnableVariable

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype aggregated in the role SwcInternalBehavior.implicitInterRunnableVariable shall be standard.⌋

[constr_2039] swImplPolicy for VariableDataPrototype in the role explicitInterRunnableVariable

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype aggregated in the role SwcInternalBehavior.explicitInterRunnableVariable shall be standard.⌋

[constr_2040] swImplPolicy for VariableDataPrototype in the role arTypedPerInstanceMemory

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype aggregated in the role SwcInternalBehavior.arTypedPerInstanceMemory shall be standard or measurementPoint.⌋

[constr_2041] swImplPolicy for VariableDataPrototype in the role staticMemory

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a VariableDataPrototype aggregated in the role InternalBehavior.staticMemory shall be standard or measurementPoint.⌋

[constr_2042] swImplPolicy for ParameterDataPrototype in ParameterInterface

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a ParameterDataPrototype owned by a ParameterInterface shall be either standard, const, or fixed.⌋

[constr_2043] swImplPolicy for ParameterDataPrototype in the role romBlock

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a ParameterDataPrototype aggregated in the role NvBlockDescriptor.romBlock shall be standard.⌋

[constr_2044] swImplPolicy for ParameterDataPrototype in the role sharedParameter

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a ParameterDataPrototype aggregated in the role SwcInternalBehavior.sharedParameter shall be standard or const.⌋

[constr_2045] swImplPolicy for ParameterDataPrototype in the role perInstanceParameter

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a ParameterDataPrototype in the role SwcInternalBehavior.perInstanceParameter shall be standard or const.⌋

[constr_2046] swImplPolicy for ParameterDataPrototype in the role constantMemory

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of a ParameterDataPrototype aggregated in the role InternalBehavior.constantMemory shall be standard, const, or fixed.⌋

[constr_2047] swImplPolicy for ArgumentDataPrototype

Imposition time: CP: IT_CpgExe

⌈The overriding value of attribute swImplPolicy of an ArgumentDataPrototype shall be standard.⌋

[constr_2049] Different ModeDeclarationGroups shall have different shortNames.

Imposition time: CP: IT_CpgExe

⌈A software component is not allowed to type multiple PortPrototypes with ModeSwitchInterfaces where the contained ModeDeclarationGroupPrototypes reference ModeDeclarationGroups with identical shortNames but different ModeDeclarations.⌋

[constr_2050] Mandatory information of a SwAxisCont

Imposition time: CP: IT_CpgExe

⌈If the attribute swAxisCont is defined for an ApplicationValueSpecification the SwAxisCont shall define

- one swAxisIndex value and
- one swArraysize value

per dimension, even in the case when the owning ApplicationValueSpecification defines only the content of a single dimensional object of (for example) category CURVE.⌋

[constr_2052] Values of swArraysize and the number of values provided by swValuesPhys shall be consistent.

Imposition time: CP: IT_ValSpec, AP: IT_ValSpec

⌈swValuesPhys shall define as many values as the attribute swArraysize (if this attribute exists) defines.

In other words, in the bound model the number of descendants (v, or vf, or vt, or vtf) shall be identical to the number of elements of the related DataPrototype typed by an ApplicationPrimitiveDataType.

If several swArraysize values are provided, the values shall be multiplied in order to get the total number of swValuesPhys values.⌋

[constr_2053] Consistency between role IUMPRNumerator and ObdRatioServiceNeeds.connectionType

Imposition time: CP: IT_RteGen

⌈If a SwcServiceDependency with a ObdRatioServiceNeeds is defined and the attribute connectionType of the contained ObdRatioServiceNeeds is set to ObdRatioConnectionKindEnum.apiUse, a RoleBasedPortAssignment with the role value IUMPRNumerator shall be defined.

If the attribute connectionType of the contained ObdRatioServiceNeeds is set to ObdRatioConnectionKindEnum.observer, the role value IUMPRNumerator is not applicable.⌋

[constr_2054] Valid targets of rptSystem

Imposition time: CP: IT_RteGen

⌈The System referenced in the role rptSystem shall be of category RPT_SYSTEM.⌋

[constr_2055] Valid targets of byPassPoint and rptHook reference, depending on the value of attribute category

Imposition time: CP: IT_RteGen

⌈

Category	*Meaning*	*Specific properties*
SW_COMPONENT_PROTOTYPE	Adds one SwComponentPrototype to an Rapid Prototyping Scenario.	The byPassPoint and rptArHook shall reference a SwComponentPrototypes.
DATA_PROTOTYPE	Adds one instance of a DataPrototype to an Rapid Prototyping Scenario.	The byPassPoint and rptArHook shall reference a DataPrototype instances in PortPrototypes.
RUNNABLE_ENTITY	Adds one RunnableEntity to an Rapid Prototyping Scenario.	The byPassPoint and rptArHook shall reference a RunnableEntity instances.
ACCESS_POINTS	Adds one VariableAccess, ParameterAccess, ServerCallPoint, AsynchronousServerCallResultPoint, InternalTriggeringPoint, ModeSwitchPoint, ModeAccessPoint or ExternalTriggeringPoint to a Rapid Prototyping Scenario.	The byPassPoint and rptArHook shall reference a VariableAccess, ParameterAccess, ServerCallPoint, AsynchronousServerCallResultPoint, InternalTriggeringPoint, ModeSwitchPoint, ModeAccessPoint or ExternalTriggeringPoint instances.

⌋

[constr_2056] Consistency of RapidPrototypingScenario with respect to rptSystem and rptArHook references

Imposition time: CP: IT_RteGen

⌈Within one RapidPrototypingScenario all rptSystem references shall point to instances in one and only one System, and if existent, all rptArHook shall point to instances in one other and only one other System.⌋

[constr_2057] Mandatory information of a RuleBasedAxisCont

Imposition time: CP: IT_CpgExe

⌈If the attribute swAxisCont is defined for an ApplicationRuleBasedValueSpecification the RuleBasedAxisCont shall define one swAxisIndex value and one swArraysize value per dimension, even in the case when the owning ApplicationRuleBasedValueSpecification defines only the content of a single dimensional object like a CURVE.⌋

[constr_2058] Mandatory information of a RuleBasedValueCont

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈If the attribute swValueCont is defined for an ApplicationRuleBasedValueSpecification the RuleBasedValueCont shall always define the attribute swAr-

raysize if the ApplicationRuleBasedValueSpecification is of category CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, or VAL_BLK.⌋

[constr_2535] Target of an autosarParameter in AutosarParameterRef shall refer to a parameter

Imposition time: CP: IT_CpgExe

⌈Except for the specifically described cases where [constr_1173] applies, the target of autosarParameter (which in fact is an instance ref) in AutosarParameterRef shall either be or be nested in ParameterDataPrototype. This means that the target shall either be a ParameterDataPrototype or an ApplicationCompositeElementDataPrototype that in turn is owned by a ParameterDataPrototype.⌋

[constr_2536] Target of an autosarVariable in AutosarVariableRef shall refer to a variable

Imposition time: CP: IT_CpgExe

⌈The target of autosarVariable (which in fact is an instance ref) in AutosarVariableRef shall either be or be nested in VariableDataPrototype. This means that the target shall either be a VariableDataPrototype or an ApplicationCompositeElementDataPrototype that in turn is owned by a VariableDataPrototype.⌋

[constr_2544] Limits need to be consistent

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈

- The limits of ApplicationDataType shall be inside the definition range of the CompuMethod

  The CompuMethod needs to be applicable for limits of an ApplicationDataType. The reason is that the internal representation of the limits of the ApplicationDataType is calculated by applying the CompuMethod.

- The internal limits defined in this way for the ApplicationDataType shall be within or equal to the internalConstrs of the mapped ImplementationDataType.

- The limits of the ImplementationDataType shall be within or equal to the limits defined by the size of the BaseType.

⌋

[constr_2545] invalidValue shall fit in the specified ranges

Imposition time: CP: IT_CpgExe

⌈The invalidValue shall be in the range of the ImplementationDataType.⌋

[constr_2549] Units of input axis shall be consistent

Imposition time: CP: IT_CpgExe

⌈All Units specified in the context of one input axis according to [TPS_SWCT_01502] shall be compatible.⌋

[constr_2550] Units of value axis shall be consistent

Imposition time: CP: IT_CpgExe

⌈All Units specified in the context of one value axis according to [TPS_SWCT_01497] shall be the same.⌋

[constr_2561] Application of DataConstrRule.constrLevel

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈DataConstrRule.constrLevel is limited to

0: This represents so called “hard limits”. They shall always be specified.

1: This represents so called “soft limits”. Soft limits may be violated after confirmation by the user of an MCD-System.

Other values may exist, but the semantics is outside the AUTOSAR scope.⌋

[constr_3688] Existence of attribute RoleBasedDataAssignment.usedDataElement.localVariable for RoleBasedDataAssignment.role = ServerServiceOffer

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value ServerServiceOffer, then the reference RoleBasedDataAssignment.usedDataElement.localVariable shall not exist.⌋

[constr_3689] Existence of attribute RoleBasedDataAssignment.usedDataElement.localVariable for RoleBasedDataAssignment.role = ClientEventSubscription

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value ClientEventSubscription, then the reference RoleBasedDataAssignment.usedDataElement.localVariable shall not exist.⌋

[constr_4002] Unambiguous mapping of modes to data types

Imposition time: CP: IT_CpgExe

⌈Within one DataTypeMappingSet, a ModeDeclarationGroup shall not be mapped to different ImplementationDataTypes.⌋

[constr_4003] Semantics of SwcModeSwitchEvent

Imposition time: CP: IT_RteGen

⌈If the value of SwcModeSwitchEvent.activation is onTransition, then SwcModeSwitchEvent shall refer to two different ModeDeclarations belonging to the same instance of ModeDeclarationGroup.

Their order defines the direction of the transition from one mode into another. In all other cases SwcModeSwitchEvent shall refer to exactly one ModeDeclaration.⌋

[constr_4004] Context of SenderReceiverAnnotation

Imposition time: CP: IT_CpgExe

⌈A SenderReceiverAnnotation shall only be aggregated by a PortPrototype typed by a SenderReceiverInterface.⌋

[constr_4005] Context of ClientServerAnnotation

Imposition time: CP: IT_CpgExe

⌈A ClientServerAnnotation shall only be aggregated by a PortPrototype typed by a ClientServerInterface.⌋

[constr_4006] Context of ParameterPortAnnotation

Imposition time: CP: IT_CpgExe

⌈A ParameterPortAnnotation shall only be aggregated by a PPortPrototype owned by a ParameterSwComponentType.⌋

[constr_4007] Context of ModePortAnnotation

Imposition time: CP: IT_CpgExe

⌈A ModePortAnnotation shall only be aggregated by a PortPrototype typed by a ModeSwitchInterface.⌋

[constr_4008] Context of TriggerPortAnnotation

Imposition time: CP: IT_CpgExe

⌈A TriggerPortAnnotation shall only be aggregated by a PortPrototype typed by a TriggerInterface.⌋

[constr_4009] Context of NvDataPortAnnotation

Imposition time: CP: IT_CpgExe

⌈An NvDataPortAnnotation shall only be aggregated by a PortPrototype typed by an NvDataInterface.⌋

[constr_4010] Context of DelegatedPortAnnotation

Imposition time: CP: IT_CompSwcT

⌈A DelegatedPortAnnotation shall only be aggregated by a PortPrototype aggregated by a CompositionSwComponentType.⌋

[constr_4012] Timeout of ModeSwitchedAckEvent

Imposition time: CP: IT_RteGen

⌈The timeout value of a WaitPoint associated with a ModeSwitchedAckEvent shall be equal to the corresponding ModeSwitchedAckRequest.timeout.⌋

[constr_4082] RunnableEntity.reentrancyLevel shall not be set.

Imposition time: CP: IT_CpgExe

⌈The optional attribute reentrancyLevel shall not be set for a RunnableEntity. This attribute would define more specific reentrancy features than the mandatory attribute canBeInvokedConcurrently. These features are currently only supported for Basic Software.⌋

[constr_5234] Existence of attribute E2EProfileCompatibilityProps.transitToInvalidExtended is mandatory for each EndToEndTransformationComSpecProps

Imposition time: CP: IT_RteGen

⌈For each EndToEndTransformationComSpecProps, a reference in the role e2eProfileCompatibilityProps to meta-class E2EProfileCompatibilityProps shall exist and the referenced E2EProfileCompatibilityProps shall define a value for the attribute transitToInvalidExtended.⌋

[constr_10005] Existence of attribute NotAvailableValueSpecification.defaultPattern

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each NotAvailableValueSpecification, attribute defaultPattern shall exist.⌋

[constr_10006] Valid interval of attribute NotAvailableValueSpecification.defaultPattern

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈The valid interval for attribute NotAvailableValueSpecification.defaultPattern is 0..255.⌋

[constr_10009] Aggregation of ApplicationRuleBasedValueSpecification

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈Each ArrayValueSpecification shall only aggregate at most one ApplicationRuleBasedValueSpecification in the role element.

If one ApplicationRuleBasedValueSpecification is aggregated then it shall be the only aggregated element, i.e. no further ValueSpecification shall exist in the same aggregation where an ApplicationRuleBasedValueSpecification is aggregated.⌋

[constr_10016] Applicability of `OsTaskExecutionEvent`

Imposition time: CP: `IT_CpgExe`

⌈An `OsTaskExecutionEvent` is only applicable for a `SwcInternalBehavior` in the context of a `ComplexDeviceDriverSwComponentType`, `EcuAbstractionSwComponentType`, or `ServiceSwComponentType`.⌋

[constr_10018] Existence of attribute `SwAxisCont.swAxisIndex`

Imposition time: CP: `IT_CpgExe`

⌈For each `SwAxisCont`, attribute `swAxisIndex` shall exist.⌋

[constr_10019] Existence of attribute `SwAxisCont.swValuesPhys`

Imposition time: CP: `IT_CpgExe`

⌈For each `SwAxisCont`, attribute `swValuesPhys` shall exist.⌋

[constr_10020] Existence of attribute `RoleBasedDataTypeAssignment.usedImplementationDataType`

Imposition time: CP: `IT_RteGen`

⌈For each `RoleBasedDataTypeAssignment`, attribute `usedImplementationDataType` shall exist.⌋

[constr_10032] Restrictions for the usage of `ServiceDependency.diagnosticRelevance`

Imposition time: CP: `IT_RteGen`

⌈The attribute `ServiceDependency.diagnosticRelevance` shall only be used for a `SwcServiceDependency` that aggregates a `BswMgrNeeds`.⌋

[constr_10033] Existence of `MemorySection.swAddrmethod`

Imposition time: CP: `IT_CpgExe`

⌈For each `MemorySection`, attribute `swAddrmethod` shall exist.⌋

[constr_10034] Existence of `MemorySection.alignment`

Imposition time: CP: `IT_CpgExe`

⌈For each `MemorySection`, attribute `alignment` shall exist if the attribute `MemorySection.swAddrmethod.memoryAllocationKeywordPolicy` is set to `MemoryAllocationKeywordPolicyType.addrMethodShortNameAndAlignment`.⌋

[constr_10040] Value of `ApplicationValueSpecification.swAxisCont.category`

Imposition time: CP: `IT_CpgExe`

⌈The value of attribute `ApplicationValueSpecification.swAxisCont.category` shall not be set to `fixAXIS`.⌋

[constr_10041] Value of role ApplicationRuleBasedValueSpecification.swAxisCont.category

Imposition time: CP: IT_CpgExe

⌈The value of attribute ApplicationRuleBasedValueSpecification.swAxisCont.category shall not be set to fixAXIS⌋

[constr_10067] Creation of AssemblySwConnector for service communication

Imposition time: CP: IT_RteGen

⌈If an AssemblySwConnector is created between two PortPrototypes and the affected PortInterfaces set the attribute isService to the value true, then at least one of the SwComponentPrototypes shall be typed by a ServiceSwComponentType.⌋

[constr_10068] Standardized values for SectionInitializationPolicyType

Imposition time: CP: IT_CpgExe

⌈The following values for SectionInitializationPolicyType are reserved by the AUTOSAR standard:

INIT To be used for (explicitly or not explicitly) initialized variables.

CLEARED To be used for not explicitly initialized variables.

POWER-ON-CLEARED To be used for variables that are not explicitly initialized (cleared) during normal start-up. Instead these are cleared only after power on reset.

⌋

[constr_10071] Allowed multiplicities of SenderComSpec attributes for communication between ApplicationSwComponentType and NvBlockSwComponentType

Imposition time: CP: IT_RteGen

⌈

Sender	ApplicationSwComponentType	
Receiver	NvBlockSwComponentType	
Queuing Configuration	non-queued	queued
SenderComSpec.transmissionAcknowledge	d/c	
SenderComSpec.dataElement	1	
SenderComSpec.handleOutOfRange	d/c	
SenderComSpec.usesEndToEndProtection	d/c	
SenderComSpec.transmissionProps.dataUpdatePeriod	0..1	
SenderComSpec.transmissionProps.minimumSendInterval	0..1	
SenderComSpec.transmissionProps.transmissionMode	0..1	
SenderComSpec.networkRepresentation	d/c	
SenderComSpec.compositeNetworkRepresentation	d/c	

▽

△

Sender	ApplicationSwComponentType	
Receiver	NvBlockSwComponentType	
Queuing Configuration	non-queued	queued
NonqueuedSenderComSpec.dataFilter	d/c	
NonqueuedSenderComSpec.initValue	0..1	

⌋

[constr_10072] Allowed multiplicities of SenderComSpec attributes for communication between NvBlockSwComponentType and ApplicationSwComponentType

Imposition time: CP: IT_RteGen

⌈

Sender	NvBlockSwComponentType	
Receiver	ApplicationSwComponentType	
Queuing Configuration	non-queued	queued
ReceiverComSpec.replaceWith	0	
ReceiverComSpec.dataElement	1	
ReceiverComSpec.receptionProps.dataUpdatePeriod	0	
ReceiverComSpec.receptionProps.timeout	0	
ReceiverComSpec.receptionProps.comHandlerTaskMapping-Enabled	0	
ReceiverComSpec.receptionProps.invalidValueBitfield-ErrorsEnabled	0	
ReceiverComSpec.receptionProps.outOfRangeBitfield-ErrorsEnabled	0	
ReceiverComSpec.handleOutOfRange	0	
ReceiverComSpec.handleOutOfRangeStatus	0	
ReceiverComSpec.transformationComSpecProps	0	
ReceiverComSpec.networkRepresentation	0	
ReceiverComSpec.compositeNetworkRepresentation	0	
QueuedReceiverComSpec.queueLength		
NonqueuedReceiverComSpec.filter	0	
NonqueuedReceiverComSpec.timeoutSubstitutionValue	0	
NonqueuedReceiverComSpec.initValue	0..1	
NonqueuedReceiverComSpec.aliveTimeout	0	
NonqueuedReceiverComSpec.enableUpdate	0	
NonqueuedReceiverComSpec.handleDataStatus	0	
NonqueuedReceiverComSpec.handleNeverReceived	0..1	
NonqueuedReceiverComSpec.handleTimeoutType	0	
NonqueuedReceiverComSpec.returnNoNewDataEnabled	0	
NonqueuedReceiverComSpec.transportErrorCountEnabled	0	
NonqueuedReceiverComSpec.valueErrorCountEnabled	0	

⌋

[constr_10073] Existence of DataReceiveErrorEvent

Imposition time: CP: IT_CpgExe

⌈A DataReceiveErrorEvent shall only exist if it refers to a given VariableDataPrototype in the role data where either

- the VariableDataPrototype is referenced from a NonqueuedReceiverComSpec in the role dataElement and the attribute aliveTimeout of the NonqueuedReceiverComSpec exists and is set to a value > 0 **or**
- the VariableDataPrototype is aggregated by a SenderReceiverInterface where attribute invalidationPolicy.handleInvalid exists and is set to the value keep.

⌋

[constr_10074] Consistency of the attribute NvBlockDescriptor.writingStrategy.role set to storeOnChange

Imposition time: CP: IT_RteGen

⌈The existence of NvBlockDescriptor.writingStrategy where attribute role is set to storeOnChange is only supported if NvBlockDescriptor.nvBlockNeeds.storeOnChange exists and is set to true.⌋

[constr_10075] Existence of CompositeRuleBasedValueSpecification.argument vs. compoundPrimitiveArgument

Imposition time: CP: IT_CpgExe

⌈For every CompositeRuleBasedValueSpecification, at most one of the aggregations

- argument
- compoundPrimitiveArgument

⌋

[constr_10087] Restriction for the existence of a SubElementMapping

Imposition time: CP: IT_RteGen

⌈The existence of a DataPrototypeMapping.subElementMapping is only supported if the PortPrototypes that are referenced by the respective SwConnector are typed by a DataInterface.⌋

[constr_10096] Shared axis shall not be a fixed axis

Imposition time: CP: IT_CpgExe

⌈An ApplicationPrimitiveDataType of category COM_AXIS shall not contain the definition of an SwCalprmAxis of category FIX_AXIS.⌋

[constr_10097] Buffer locking is only supported if `returnValueProvision` is set to `returnValueProvided`

Imposition time: CP: `IT_CpgExe`

⌈Setting the value of attribute `PortAPIOption.supportedFeature.supportBufferLocking` to value `supportsBufferLocking` is only supported if the `AbstractAccessPoint`s that refer to values in the respective `PortAPIOption.port` do **not** define `AbstractAccessPoint.returnValueProvision` **or** set the value of `AbstractAccessPoint.returnValueProvision` to `returnValueProvided`.⌋

[constr_10099] Allowed values of the attribute `SwDataDefProps.swImplPolicy` vs. `DataPrototype`s and their roles

Imposition time: CP: `IT_CpgExe`

⌈

Attribute of `SwImplPolicyEnum`	`VariableDataPrototype`							`ParameterDataPrototype`					Misc.
	`VariableDataPrototype` in `SenderReceiverInterface`	`VariableDataPrototype` in `NvDataInterface`	`VariableDataPrototype` in role `ramBlock`	`VariableDataPrototype` in role `implicitInterRunnableVariable`	`VariableDataPrototype` in role `explicitInterRunnableVariable`	`VariableDataPrototype` in role `argTypedPerInstanceMemory`	`VariableDataPrototype` in role	`ParameterDataPrototype` in	`ParameterDataPrototype` in role	`ParameterDataPrototype` in role	`ParameterDataPrototype` in role	`ParameterDataPrototype` in role	`ConstantMemoryPrototype` in role
`const`								x		x	x	x	
`fixed`								x				x	
`measurementPoint`	x					x	x						
`queued`	x												
`standard`	x	x	x	x	x	x	x	x	x	x	x	x	x

⌋

[constr_10104] `RoleBasedPortAssignment` where attribute `role` is set to `CallbackGetFaultDetectCounter` shall refer to a `PPortPrototype` in the role `portPrototype`

Imposition time: CP: `IT_RteGen`

⌈If a `SwcServiceDependency` aggregates both

- a `DiagnosticEventNeeds` that in turn aggregates `DiagEventDebounceMonitorInternal` in the role `diagEventDebounceAlgorithm` and
- a `RoleBasedPortAssignment` in the role `assignedPort` where attribute `role` is set to `CallbackGetFaultDetectCounter`,

then the target of the reference `SwcServiceDependency.assignedPort.portPrototype` shall be a `PPortPrototype`.⌋

[constr_10118] Structural consistency of the modeling of `InvalidationPolicy`

Imposition time: CP: `IT_CpgExe`

⌈A `dataElement` referenced by an `InvalidationPolicy` shall be owned by the `SenderReceiverInterface` that also owns the `InvalidationPolicy`.⌋

[constr_10119] `SenderReceiverInterface`.dataElement shall be referenced by at most one `InvalidationPolicy`

Imposition time: CP: `IT_CpgExe`

⌈Any `SenderReceiverInterface.dataElement` shall be referenced by at most one `InvalidationPolicy` in the role `InvalidationPolicy.dataElement`.⌋

[constr_10120] Structural consistency of the modeling of `MetaDataItemSet`

Imposition time: CP: `IT_CpgExe`

⌈A `dataElement` referenced by an `MetaDataItemSet` in the role `dataElement` shall be owned by the `SenderReceiverInterface` that also owns the `MetaDataItemSet`.⌋

[constr_10121] `SenderReceiverInterface.dataElement` shall be referenced by at most one `MetaDataItemSet`

Imposition time: CP: `IT_CpgExe`

⌈Any `SenderReceiverInterface.dataElement` shall be referenced by at most one `MetaDataItemSet` in the role `MetaDataItemSet.dataElement`.⌋

[constr_10123] Existence of attribute `DtcStatusChangeNotificationNeeds.notificationTime`

Imposition time: CP: `IT_RteGen`

⌈Attribute `DtcStatusChangeNotificationNeeds.notificationTime` shall only exist if the enclosing `SwcServiceDependency` contains a `RoleBasedPortAssignment` where attribute role is set to the value `ClearDtcNotification`.⌋

[constr_10196] Definition of `invalidValue` for `DataPrototype` typed by an `ImplementationDataType` that references a `CompuMethod` of `category` `TEXTTABLE` or `BITFIELD_TEXTTABLE`

Imposition time: CP: `IT_CpgExe`

⌈If an `invalidValue` is defined for a `DataPrototype` that is typed by an `ImplementationDataType` that references or inherits (see [constr_1015]) a `CompuMethod` of `category` `TEXTTABLE` or `BITFIELD_TEXTTABLE`, the applicable `ValueSpecification` shall be a `TextValueSpecification` **if** the value fits into the intervals defined by the `CompuMethod`.

In this case the value provided shall match one of the applicable text values (`vt`, `shortLabel`, `symbol`) defined by the applicable `CompuScale`s.⌋

[constr_10372] Relation between *Type of* `PortPrototype`, *Type of* `ComSpec`, and *Type of* `PortInterface`

Imposition time: CP: `IT_CpgExe`

⌈With respect to [constr_1043], if a *Type of* `PortPrototype` aggregates a *Type of* `ComSpec`, then the *Type of* `PortPrototype` shall

- reference a *Type of* `PortInterface` in the role *Role of Type-Ref* and
- the *Role of Element* that is referenced from the *Type of* `ComSpec` shall be aggregated by the exact same *Type of* `PortInterface` that is also referenced by the enclosing *Type of* `PortPrototype` in the role *Role of Type-Ref*.

⌋

[constr_10373] `ImplementationDataType` of `category` `VALUE` shall not refer to `SwBaseType` of `category` `VOID`

Imposition time: CP: `IT_CpgExe`

⌈An `ImplementationDataType` where attribute `category` is set to `VALUE` shall not reference (in the role `swDataDefProps.baseType`) a `SwBaseType` where attribute `category` is set to `VOID`.⌋

[constr_10383] Supported value encodings for `SwBaseType` in the context of `PortInterface`s where attribute `isService` is set to `false`

Imposition time: CP: `IT_CpgExe`

⌈The supported value encodings for the usage within a `PortInterface` where attribute `isService` is set to `false` are:

- `2C`: Two's complement
- `IEEE754`: floating-point numbers
- `ISO-8859-1`: single-byte coded character
- `ISO-8859-2`: single-byte coded character
- `WINDOWS-1252`: single-byte coded character
- `UTF-8`: UCS Transformation Format 8
- `UTF-16`: Character encoding for Unicode code points based on 16 bit code units, see [7, ISO 10646]
- `UCS-2`: Universal Character Set 2
- `NONE`: Unsigned Integer
- `BOOLEAN`: This represents an integer to be interpreted as boolean.

⌋

[constr_10415] Initial value on the level of an ImplementationDataTypeElement where attribute isOptional is set to the value True

Imposition time: CP: IT_CpgExe

⌈The initial value used on the level of an ImplementationDataTypeElement where attribute isOptional is set to the value True shall **not** be initialized using a NotAvailableValueSpecification.⌋

[constr_10424] Reference from MemorySection to ExecutableEntity

Imposition time: CP: IT_CpgExe

⌈Each ExecutableEntity shall only be referenced by exactly one MemorySection.⌋

[constr_10433] Existence of attributes of ApplicationDataType depending on the category

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈

Attributes of ApplicationDataType	Owner				Attribute Existence per ApplicationDataType.category												
	ApplicationRecordDataType	ApplicationRecordElement	ApplicationArrayDataType	ApplicationArrayElement	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
element	x						1..*										
isOptional		x					0..1										
element			x					1									
dynamicArraySizeProfile			x					0..1									
arraySizeHandling				x				0..1									
arraySizeSemantics				x				0..1									
maxNumberOfElements				x				1									
swDataDefProps	x	x	x	x	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1

⌋

[constr_10434] Existence of attributes of ImplementationDataType depending on the category

Imposition time: CP: IT_CpgExe

⌈

Attributes of ImplementationDataType	Attribute Existence per Category						
	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
dynamicArraySizeProfile							0..1
isStructWithOptionalElement					0..1		
typeEmitter	0..1	0..1	0..1	0..1	0..1	0..1	0..1
symbolProps	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swDataDefProps	0..1	0..1	0..1	0..1	0..1	0..1	0..1
subElement					1..*	1..*	1
subElement.isOptional					0..1		
subElement.arraySize							0..1
subElement.arraySizeSemantics							0..1
subElement.arraySizeHandling							0..1
subElement.arrayImplPolicy							0..1

⌋

[constr_10435] Existence of attributes of ImplementationDataTypeElement depending on the category

Imposition time: CP: IT_CpgExe

⌈

Attributes of ImplementationDataTypeElement	Attribute Existence per Category						
	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
swDataDefProps	0..1	0..1	0..1	0..1	0..1	0..1	0..1
subElement					1..*	1..*	1
subElement.isOptional					0..1		
subElement.arraySize							0..1
subElement.arraySizeSemantics							0..1
subElement.arraySizeHandling							0..1
subElement.arrayImplPolicy							0..1

⌋

[constr_10439] Initialization of a DataPrototype typed by a Compound Primitive Data Type

Imposition time: CP: IT_CpgExe

⌈If a DataPrototype that is typed by a Compound Primitive Data Type according to [TPS_SWCT_01179] needs to be initialized by a constant value, then the initialization shall only be provided in the form of an ApplicationValueSpecification.⌋

[constr_10502] Number of elements of ApplicationValueSpecification.swValueCont.swArraysize vs. ApplicationValueSpecification.category

Imposition time: CP: IT_ValSpec, AP: IT_ValSpec

⌈

Value of category	Number of values in swValueCont.swArraysize
CURVE	1
MAP	2
CUBOID	3
CUBE_4	4
CUBE_5	5
COM_AXIS	1
RES_AXIS	1
VAL_BLK	1..*

⌋

[constr_10503] ApplicationValueSpecification where attribute category is set to MAP, CUBOID, CUBE_4, or CUBE_5 and ROW_DIR SwRecordLayout

Imposition time: CP: IT_ValSpec

⌈In the context of an ApplicationValueSpecification where attribute category is set to MAP, CUBOID, CUBE_4, or CUBE_5 that is applied to a DataPrototype typed by an ApplicationPrimitiveDataType where the swDataDefProps.swRecordLayout refers to a SwRecordLayout with a ROW_DIR approach, the value of ApplicationValueSpecification.swValueCont.swArraysize.v[*i*] (i.e., counting up from the first element of swArraysize.v) shall be identical to the number of axis points of the respective SwCalprmAxisSet.swCalprmAxis where attribute swAxisIndex is set to *i*:

- If the respective SwCalprmAxis is a SwAxisGrouped, then the number of axis points shall be retrieved from the attribute subElement.arraySize of the ImplementationDataType that is referenced by a DataTypeMap that also references the ApplicationDataType referenced in the role SwAxisGrouped.sharedAxisType.
- If the respective SwCalprmAxis is a SwAxisIndividual, the number of axis points is identical to the value of attribute SwAxisIndividual.swMaxAxisPoints.

⌋

[constr_10504] ApplicationValueSpecification where attribute category is set to VAL_BLK and ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSizeMult exists for ROW_DIR SwRecordLayout

Imposition time: CP: IT_ValSpec

⌈If the attribute ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSizeMult exists, then the value of ApplicationValueSpecification.swValueCont.swArraysize can be identical to the value of ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSizeMult if the referenced ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout defines a row-first ROW_DIR.⌋

[constr_10505] ApplicationValueSpecification where attribute category is set to MAP, CUBOID, CUBE_4, or CUBE_5 and COLUMN_DIR SwRecordLayout

Imposition time: CP: IT_ValSpec

⌈In the context of an ApplicationValueSpecification where attribute category is set to MAP, CUBOID, CUBE_4, or CUBE_5 that is applied to a DataPrototype typed by an ApplicationPrimitiveDataType where the swDataDefProps.swRecordLayout refers to a SwRecordLayout with a COLUMN_DIR approach, the value of ApplicationValueSpecification.swValueCont.swArraysize.v[-*i*] (i.e., counting down from the last element of swArraysize.v) shall be identical to the number of axis points of the respective SwCalprmAxisSet.swCalprmAxis where attribute swAxisIndex is set to *i*:

- If the respective SwCalprmAxis is a SwAxisGrouped, then the number of axis points shall be retrieved from the attribute subElement.arraySize of the ImplementationDataType that is referenced by a DataTypeMap that also references the ApplicationDataType referenced in the role SwAxisGrouped.sharedAxisType.
- If the respective SwCalprmAxis is a SwAxisIndividual, the number of axis points is identical to the value of attribute SwAxisIndividual.swMaxAxisPoints.

⌋

[constr_10506] ApplicationValueSpecification where attribute category is set to VAL_BLK and ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSizeMult exists for COLUMN_DIR SwRecordLayout

Imposition time: CP: IT_ValSpec

⌈If the attribute ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSizeMult exists, then the value of attribute ApplicationValueSpecification.swValueCont.swArraysize can be taken over from the reversed values of ApplicationPrimitiveDataType.swDataDefProps.swValue-

BlockSizeMult if the referenced ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout defines a column-first COLUMN_DIR layout.⌋

[constr_10507] ApplicationValueSpecification where attribute category is set to VAL_BLK and ApplicationPrimitiveDataType.swDataDefProps.swValueBlockSize exists

Imposition time: CP: IT_ValSpec, AP: IT_ValSpec

⌈If the attribute AutosarDataType.swDataDefProps.swValueBlockSize exists, then the value of ApplicationValueSpecification.swValueCont.swArraysize shall contain a single v and the value of v shall be identical to the value of attribute swValueBlockSize.⌋

[constr_10520] Multiplicity of AssemblySwConnector.provider

Imposition time: CP: IT_CompSwcT

⌈For each AssemblySwConnector, the reference AssemblySwConnector.provider shall exist.⌋

[constr_10521] Multiplicity of AssemblySwConnector.requester

Imposition time: CP: IT_CompSwcT

⌈For each AssemblySwConnector, the reference AssemblySwConnector.requester shall exist.⌋

[constr_10525] Existence of attribute ApplicationValueSpecification.category

Imposition time: CP: IT_CpgExe, AP: IT_BefAraApiGen

⌈For each ApplicationValueSpecification, attribute category shall exist.⌋

[constr_10527] Existence of RoleBasedDataAssignment.usedDataElement.autosarVariable for RoleBasedDataAssignment.role = ramBlock

Imposition time: CP: IT_RteGen

⌈If the attribute RoleBasedDataAssignment.role is set to the value ramBlock, then the reference RoleBasedDataAssignment.usedDataElement.autosarVariable shall not exist.⌋

[constr_10529] Existence of AsynchronousServerCallResultPoint for AsynchronousServerCallPoint where attribute timeout is defined

Imposition time: CP: IT_CpgExe

⌈For each AsynchronousServerCallPoint where attribute timeout exists, an AsynchronousServerCallResultPoint shall exist that references the AsynchronousServerCallPoint in the role asynchronousServerCallPoint.⌋

[constr_10532] Restriction for the aggregation in the role SenderComSpec.transmissionProps.onChangeDataPrototype

Imposition time: CP: IT_CpgExe

⌈If the SenderComSpec.dataElement refers to an AutosarDataPrototype aggregated by a SenderReceiverInterface in the role dataElement, then a DataPrototypeReference aggregated in the role SenderComSpec.transmissionProps.onChangeDataPrototype shall only exist as

- a DataPrototypeInPortInterfaceRef that aggregates a DataPrototypeInSenderReceiverInterfaceInstanceRef in the role dataPrototypeInSenderReceiverInterface or
- an ImplementationDataTypeElementInPortInterfaceRef.

⌋

[constr_10533] Existence of the reference in the role TransmissionComSpecProps.onChangeDataPrototype.dataPrototypeInSenderReceiverInterface.rootDataPrototypeInSr

Imposition time: CP: IT_CpgExe

⌈If all of the following conditions apply:

- the SenderComSpec.dataElement refers to an AutosarDataPrototype aggregated by a SenderReceiverInterface in the role dataElement and
- the aggregation in the role TransmissionComSpecProps.onChangeDataPrototype.dataPrototypeInSenderReceiverInterface exists,

then the aggregation in the role TransmissionComSpecProps.onChangeDataPrototype.dataPrototypeInSenderReceiverInterface.rootDataPrototypeInSr shall **not** exist.⌋

[constr_10534] Existence of TransmissionComSpecProps.onChangeDataPrototype.rootDataPrototype

Imposition time: CP: IT_CpgExe

⌈If all of the following conditions apply:

- the SenderComSpec.dataElement refers to an AutosarDataPrototype aggregated by a SenderReceiverInterface in the role dataElement and
- the aggregation in the role TransmissionComSpecProps.onChangeDataPrototype exists,

then the aggregation in the role TransmissionComSpecProps.onChangeDataPrototype.rootDataPrototype shall **not** exist⌋

[constr_10538] Existence of attribute ReceiverComSpec.dataElement

Imposition time: CP: IT_CpgExe

⌈For each ReceiverComSpec, attribute dataElement shall exist.⌋

[constr_10539] Existence of attribute `SenderComSpec.dataElement`

Imposition time: CP: `IT_CpgExe`, AP: `IT_BefAraApiGen`

⌈For each `SenderComSpec`, attribute `dataElement` shall exist.⌋

[constr_10542] `RunnableEntity` is referenced by an `OperationInvokedEvent`

Imposition time: CP: `IT_CpgExe`

⌈A `RunnableEntity` that is referenced by one or more (according to [TPS_SWCT_01225]) `OperationInvokedEvent`s in the role `startOnEvent` shall **not** be referenced in the same role (`startOnEvent`) by any other subclass of `RTEEvent`.⌋

[constr_10543] Uniqueness of reference `PortAPIOption.port`

Imposition time: CP: `IT_CpgExe`

⌈Any `PortPrototype` may be referenced **at most once** in the role `PortAPIOption.port`.⌋

[constr_10544] Ownership of reference `PortAPIOption.port`

Imposition time: CP: `IT_CpgExe`

⌈A `PortPrototype` referenced in the role `PortAPIOption.port` shall be owned by the same `AtomicSwComponentType` that also owns the `SwcInternalBehavior` that in turn owns the `PortAPIOption` from which the `PortPrototype` is referenced.⌋

[constr_10558] `SwBaseType` associated with corresponding `ApplicationRecordElement` and `ImplementationDataTypeElement`

Imposition time: CP: `IT_CpgExe`

⌈If

- an `ApplicationRecordElement` is implicitly (i.e. by position in the enclosing `ApplicationRecordDataType`) mapped to an `ImplementationDataTypeElement` of `category` `VALUE` and
- the `ApplicationRecordElement` typed by an `ApplicationPrimitiveDataType` and a `DataTypeMap` exists that maps the `ApplicationPrimitiveDataType` to an `ImplementationDataType` of `category` `VALUE`,

then the affected `ImplementationDataType` of `category` `VALUE` shall reference the **identical** `SwBaseType` as the affected `ImplementationDataTypeElement`.⌋

[constr_10559] Uniqueness of `DataPrototypeMapping.firstDataPrototype` and `secondDataPrototype`

Imposition time: CP: `IT_RteGen`, AP: `IT_BefAraApiGen`

⌈Within the context of a `VariableAndParameterInterfaceMapping`, **no two** `DataPrototypeMapping`s shall exist where the targets of the combination of ref-

erences in the roles firstDataPrototype and secondDataPrototype are identical.⌋

[constr_10560] Uniqueness of ClientServerOperationMapping.firstOperation and secondOperation

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈Within the context of a ClientServerInterfaceMapping, **no two** ClientServerOperationMappings shall exist where the targets of the combination of references in the roles firstOperation and secondOperation are identical.⌋

[constr_10561] Uniqueness of ClientServerApplicationErrorMapping.firstApplicationError and secondApplicationError

Imposition time: CP: IT_RteGen, AP: IT_BefAraApiGen

⌈Within the context of a ClientServerInterfaceMapping, **no two** ClientServerApplicationErrorMappings shall exist where the targets of the combination of references in the roles firstApplicationError and secondApplicationError are identical.⌋

[constr_10562] Uniqueness of ModeDeclarationGroupPrototypeMapping.firstModeGroup and secondModeGroup

Imposition time: CP: IT_RteGen

⌈Within the context of a ModeInterfaceMapping, **no two** ModeDeclarationGroupPrototypeMappings shall exist where the targets of the combination of references in the roles firstModeGroup and secondModeGroup are identical.⌋

[constr_10563] Uniqueness of ModeDeclarationMapping.firstMode and secondMode

Imposition time: CP: IT_RteGen

⌈Within the context of a ModeDeclarationMappingSet, no two ModeDeclarationMappings shall exist where the targets of the combination of references in the roles firstMode and secondMode are identical.⌋

[constr_10564] Uniqueness of TriggerMapping.firstTrigger and secondTrigger

Imposition time: CP: IT_RteGen

⌈Within the context of a TriggerInterfaceMapping, **no two** TriggerMappings shall exist where the targets of the combination of references in the roles firstTrigger and secondTrigger are identical.⌋

[constr_10565] Uniqueness of SubElementMapping.firstElement and secondElement

Imposition time: CP: IT_RteGen

⌈Within the context of a DataPrototypeMapping, **no two** SubElementMappings shall exist where the targets of the combination of references in the roles firstElement and secondElement are identical.⌋

[constr_10575] No multiple instantiation of NvBlockSwComponentType

Imposition time: CP: IT_RteGen

⌈For each NvBlockSwComponentType, attribute internalBehavior.supportsMultipleInstantiation shall **always** be set to false.⌋

[constr_10606] Existence of ConstantSpecificationMapping or CalibrationParameterValue for ApplicationValueSpecification or ApplicationRuleBasedValueSpecification of category COM_AXIS, STD_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, and CUBE_5

Imposition time: CP: IT_RteGen

⌈Any ConstantSpecification that aggregates an ApplicationValueSpecification or ApplicationRuleBasedValueSpecification of category COM_AXIS, STD_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, and CUBE_5 shall be referenced from

- a ConstantSpecificationMapping in the role applConstant and/or
- a CalibrationParameterValue in the role applInitValue.

⌋

[constr_10607] Number of ConstantSpecificationMappings that are allowed to reference a ApplicationValueSpecification or ApplicationRuleBasedValueSpecification in the context of an InternalBehavior

Imposition time: CP: IT_RteGen

⌈Within the collection of all ConstantSpecificationMappings owned by ConstantSpecificationMappingSets referenced by a single InternalBehavior, at most one ConstantSpecificationMapping shall refer to any given ApplicationValueSpecification or ApplicationRuleBasedValueSpecification.⌋

[constr_10608] Number of ConstantSpecificationMappings that are allowed to reference a ApplicationValueSpecification or ApplicationRuleBasedValueSpecification in the context of a ParameterSwComponentType

Imposition time: CP: IT_RteGen

⌈Within the collection of all ConstantSpecificationMappings owned by ConstantSpecificationMappingSets referenced by a single ParameterSw-

ComponentType, at most one ConstantSpecificationMapping shall refer to any given ApplicationValueSpecification or ApplicationRuleBasedValue-Specification.⌋

[constr_10610] Compatibility of PhysicalDimensions in the context is the creation of an ApplicationValueSpecification

Imposition time: CP: IT_CpgExe

⌈In the context of the creation of an ApplicationValueSpecification, two PhysicalDimension definitions are compatible if and only if the values of

- lengthExp
- massExp
- timeExp
- currentExp
- temperatureExp
- molarAmountExp
- luminousIntensityExp

are identical and **either**

- the shortNames are identical **or**
- a PhysicalDimensionMapping exists that maps one of the Physical-Dimensions in the role firstPhysicalDimension and the other Physical-Dimension in the role secondPhysicalDimension.

⌋

[constr_10632] Compatibility of SwTextProps

Imposition time: CP: IT_RteGen

⌈Two SwTextProps are compatible **if and only if** they have identical values of attributes

- arraySizeSemantics
- swMaxTextSize

and the following attributes of the reference in the role baseType are identical:

- baseTypebaseTypeDefinition.baseTypeEncoding
- baseTypebaseTypeDefinition.baseTypeSize
- baseTypebaseTypeDefinition.byteOrder

⌋

[constr_10667] Simple compatibility of `ImplementationDataType`s

Imposition time: CP: `IT_RteGen`

⌈Instances of `ImplementationDataType` are compatible if, after all type-references are resolved, all of the following rules apply:

1. They have the same `category`.
2. They have the identical structure (this refers to `ImplementationDataTypeElement` and their `subElement`s).
3. The attributes `arraySize` and `arraySizeSemantics` have (given the existence) identical values.
4. For each `ImplementationDataType.subElement`, the attribute `isOptional` shall either
   - not exist on both sides or
   - be set to the value `false` if it only exists on one side or
   - have the identical value on both sides.
5. The `swDataDefProps` (after consideration of [constr_1015]) attached to the M1 data types are compatible.

⌋

[constr_10668] Restriction for the existence of attributes of `RunnableEntityArgument` for port-defined argument values

Imposition time: CP: `IT_CpgExe`

⌈Any `RunnableEntity.argument` that corresponds (based on the order index) to a port-defined argument value shall not define any of

- the reference in the role `implementationDataType` or
- the attribute `serverArgumentImplPolicy`.

⌋

[constr_10669] Existence of reference in the role `RunnableEntityArgument.implementationDataType`

Imposition time: CP: `IT_CpgExe`

⌈For each `RunnableEntityArgument` that represents an argument of a `ClientServerOperation`, the reference in the role `implementationDataType` shall exist.⌋

[constr_10670] Compatibility between the corresponding `ClientServerOperation.argument` and `RunnableEntity.argument`

Imposition time: CP: `IT_CpgExe`

⌈If a `ClientServerOperation` is referenced in the role `operation` by an `OperationInvokedEvent` that in turn references a `RunnableEntity` in the role `startOnEvent` and if the `RunnableEntity` owns `RunnableEntityArgument`s in the role `argument`, the corresponding (i.e. same position in the ordered collection of arguments) `ClientServerOperation.argument` and `RunnableEntity.argument` shall fulfill the following conditions:

- The `ImplementationDataType` associated with the `ClientServerOperation.argument` and the `ImplementationDataType` referenced in the role `RunnableEntity.argument.implementationDataType` shall be compatible according to [constr_10667].
- The attribute `ClientServerOperation.argument.direction` and `RunnableEntity.argument.direction` shall have the same value.

⌋

[constr_10671] `ReceptionComSpecProps.comHandlerTaskMappingEnabled` vs. implicit data read access

Imposition time: `IT_CpgExe`

⌈If the `ReceiverComSpec.dataElement` is referenced by a `VariableAccess` that is aggregated by `RunnableEntity` in the `dataReadAccess` role (which translates to implicit data read access), then `ReceptionComSpecProps.comHandlerTaskMappingEnabled` shall not be set to `true`.⌋

[constr_10672] Applicability of `transportErrorCountEnabled`, `valueErrorCountEnabled`, `returnNoNewDataEnabled`

Imposition time: CP: `IT_CpgExe`

⌈The attributes

- `NonqueuedReceiverComSpec.transportErrorCountEnabled`,
- `NonqueuedReceiverComSpec.valueErrorCountEnabled`, and
- `NonqueuedReceiverComSpec.returnNoNewDataEnabled`

shall only exist if the attribute `ReceptionComSpecProps.comHandlerTaskMappingEnabled` is set to `true`.⌋

[constr_10673] Existence of `ReceptionComSpecProps.comHandlerTaskMappingEnabled` vs. `ReceptionComSpecProps.timeout`

Imposition time: CP: `IT_CpgExe`

⌈In each `ReceptionComSpecProps`, **at most one** of the attributes

- comHandlerTaskMappingEnabled (timeout monitoring by the “COM handler”) and
- timeout (timeout monitoring by the application software)

shall exist.⌋

[constr_10674] Existence of attribute ReceptionComSpecProps.outOfRange-BitfieldErrorsEnabled

Imposition time: CP: IT_CpgExe

⌈Attribute ReceptionComSpecProps.outOfRangeBitfieldErrorsEnabled shall only exist if the ReceiverComSpec.dataElement is typed by either

- an ApplicationRecordDataType or
- an ImplementationDataType where (after the removal of all indirections) attribute category is set to the value STRUCTURE.

⌋

[constr_10675] Existence of attribute ReceptionComSpecProps.invalid-ValueBitfieldErrorsEnabled

Imposition time: CP: IT_CpgExe

⌈Attribute ReceptionComSpecProps.invalidValueBitfieldErrorsEnabled shall only exist if the ReceiverComSpec.dataElement is typed by either

- an ApplicationRecordDataType or
- an ImplementationDataType where (after the removal of all indirections) attribute category is set to the value STRUCTURE.

⌋

[constr_10744] Multiplicity of the reference in the role DataPrototypeMapping.textTableMapping

Imposition time: IT_RteGen

⌈Within the context of a DataPrototypeMapping, the reference in the role text-TableMapping shall exist **at most twice**.⌋

[constr_10745] Multiplicity of the reference in the role SubElementMapping.textTableMapping

Imposition time: IT_RteGen

⌈Within the context of a SubElementMapping, the reference in the role textTable-Mapping shall exist **at most twice**.⌋

[constr_10792] Relation between baseTypeSize, baseTypeEncoding and InternalConstrs

Imposition time: CP: IT_CpgExe

⌈

baseTypeEncoding	Lowest InternalConstrs.lowerLimit	Highest InternalConstrs.upperLimit
NONE	0	$2^{baseTypeSize} - 1$
2C	$-2^{baseTypeSize-1}$	$2^{baseTypeSize-1} - 1$
IEEE754 (32 bit)	$-3.4 * 10^{38}$	$3.4 * 10^{38}$
IEEE754 (64 bit)	$-1.8 * 10^{308}$	$1.8 * 10^{308}$
DSP-FRACTIONAL	-1.0	1.0

⌋

[constr_10793] Restriction for the definition of internalConstrs in the context of an ImplementationDataType of category TYPE_REFERENCE

Imposition time: CP: IT_CpgExe

⌈For each ImplementationDataType of category TYPE_REFERENCE, the definition of

- swDataDefProps.dataConstr.dataConstrRule.internalConstrs.lowerLimit shall be **greater than or equal** to the respective lowerLimit defined in the context of the corresponding ImplementationDataType where attribute category is not set to TYPE_REFERENCE and
- swDataDefProps.dataConstr.dataConstrRule.internalConstrs.upperLimit shall be **less than or equal** to the respective upperLimit defined in the context of the corresponding ImplementationDataType where attribute category is not set to TYPE_REFERENCE.

⌋

2.6 CP_TPS_SystemTemplate

[constr_1198] TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping in case no DataTransformation is used

Imposition time: CP: IT_EcuExt, AP: IT_Mani

⌈The ISignal that is referenced by a SystemSignal that in turn is referenced by a TriggerToSignalMapping in the role systemSignal shall have the length attribute set to 0 if the ISignal does not reference a DataTransformation in the role dataTransformation.⌋

[constr_1199] ISignals relating to systemSignals eligible for a TriggerToSignalMapping shall use update bit in case no DataTransformation is used

Imposition time: CP: IT_SysDesc, AP: IT_Mani

⌈An ISignal

- that is used to reference a systemSignal that in turn is referenced by a TriggerToSignalMapping and
- does not reference a DataTransformation in the role dataTransformation

shall be referenced by an ISignalToIPduMapping where the attribute updateIndicationBitPosition is defined.⌋

[constr_1265] DoIpGidSynchronizationNeeds can only exist once per ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈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

Imposition time: CP: IT_EcuExt

⌈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

Imposition time: CP: IT_EcuExt

⌈Within the context of one System of category ECU_EXTRACT, there can only be at most one DoIpActivationLineNeeds.⌋

[constr_1367] periodicResponseUudt.periodicResponseUudt shall only refer to a DcmIPdu

Imposition time: CP: IT_SysDesc

⌈If the role periodicResponseUudt exists then every PduTriggering referenced in the role periodicResponseUudt shall only refer to a DcmIPdu.⌋

[constr_1368] Limitation of the target of references from DiagnosticConnection

Imposition time: CP: IT_SysDesc

⌈DiagnosticConnection shall only reference (via the indirection created by TpConnectionIdent) the following sub-classes of the meta-class TpConnection:

- CanTpConnection
- FlexrayTpConnection

- DoIpTpConnection

⌋

[constr_1369] CommunicationConnectors shall be attached to the same CommunicationCluster

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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.⌋

[constr_1371] Consistency of attribute host

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_1373] GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE shall not be referenced in the role GlobalTimeGateway.master.⌋

[constr_1374] Only fan-out possible for GlobalTimeGateway

Imposition time: CP: IT_SysDesc

⌈For all GlobalTimeGateways that refer to the same EcuInstance the condition applies that no two GlobalTimeGateways shall refer to the same GlobalTimeMaster.⌋

[constr_1387] Transmission of `Variable-Size Array Data Type`s by means of a Transformer

Imposition time: CP: `IT_SysDesc`

⌈If a Transformer is used for the transmission of a `Variable-Size Array Data Type`s 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_01641] and [TPS_SWCT_01642] are not supported.⌋

[constr_1441] In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_1641] Consistent assignment of TLV data ids to `ApplicationRecordDataType`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_1643] Completeness of the existence of a set of `TlvDataIdDefinition.tlvArgument`s

Imposition time: CP: `IT_SysDesc`

⌈If the reference `TlvDataIdDefinition.tlvArgument`s exists for one `argument` of a given `ClientServerOperation` then further `TlvDataIdDefinition.tlvArgument`s shall exist for all `argument`s of the given `ClientServerOperation` and all affected `TlvDataIdDefinition`s shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.⌋

[constr_1644] Completeness of the existence of a set of TlvDataIdDefinition.tlvRecordElements

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_1646] Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ArgumentDataPrototype

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈For all `TlvDataIdDefinition` that are referencing `ImplementationDataTypeElements` 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`.⌋

[constr_1649] `TlvDataIdDefinition` referencing `ArgumentDataPrototype`

Imposition time: CP: `IT_SysDesc`

⌈Each `ArgumentDataPrototype` shall be referenced at most once in the role `tlvArgument` in the context of the same `SOMEIPTransformationISignalProps`.⌋

[constr_1650] `TlvDataIdDefinition` referencing `ApplicationRecordElement`

Imposition time: CP: `IT_SysDesc`

⌈Each `ApplicationRecordElement` shall be referenced at most once in the role `tlvRecordElement` in the context of the same `SOMEIPTransformationISignalProps`.⌋

[constr_1651] `TlvDataIdDefinition` referencing `ImplementationDataTypeElement`

Imposition time: CP: `IT_SysDesc`

⌈Each `ImplementationDataTypeElement` shall be referenced at most once in the role `tlvImplementationDataTypeElement` in the context of the same `SOMEIPTransformationISignalProps`.⌋

[constr_1652] Definition of static length fields sizes in case of TLV usage

Imposition time: CP: `IT_SysDesc`

⌈If `TlvDataIdDefinitions` are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, `sizeOfStringLengthFields` and `sizeOfUnionLengthFields` shall be greater than 0.⌋

[constr_1653] Identical values for length fields sizes in case of TLV usage

Imposition time: CP: `IT_SysDesc`

⌈If `TlvDataIdDefinitions` are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, `sizeOfStringLengthFields` and `sizeOfUnionLengthFields` shall have an identical value.⌋

[constr_1654] No definition of length field sizes on DataPrototype level in case of TLV usage

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_1655] The mutual existence of LinMasters in the LinSlave EcuExtract

Imposition time: CP: IT_EcuExt

⌈A LinMaster shall not be part of the EcuExtract of a corresponding LinSlave.⌋

[constr_1656] No application-level write access to LinErrorResponse.responseError on Lin slave

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_1657] Existence of LinPhysicalChannel.scheduleTable

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_1669] Existence of PduTriggering.secOcCryptoServiceMapping

Imposition time: CP: IT_SysDesc

⌈The reference PduTriggering.secOcCryptoServiceMapping shall only exist if the PduTriggering also references a SecuredIPdu in the role iPdu.⌋

[constr_1670] Prohibition of usage of tlsCryptoMapping in case of UDP socket connections

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.

⌋

[constr_1672] Existence of TlsCryptoCipherSuite.certificate and TlsCryptoCipherSuite.pskIdentity in the *server* role

Imposition time: CP: IT_SysDesc

⌈**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.⌋

[constr_2025] Uniqueness of symbol attributes

Imposition time: CP: IT_EcuExt

⌈With the exception of RunnableEntities that are subject to [constr_1234] (Runnable Entities 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.⌋

[constr_3000] valid SenderRecCompositeTypeMappings

Imposition time: CP: IT_EcuExt

⌈All SenderRecRecordElementMappings or SenderRecArrayElementMappings aggregated in the context of a given SenderReceiverToSignalGroupMapping shall reference a SystemSignal that is also referenced in the role systemSignal by the SystemSignalGroup that is referenced by the enclosing SenderReceiverToSignalGroupMapping in the role signalGroup.⌋

[constr_3002] valid swcToImplMapping

Imposition time: CP: IT_EcuExt

⌈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⌋

[constr_3003] Number of CAN channels

Imposition time: CP: IT_SysDesc

⌈CAN clusters shall aggregate exactly one PhysicalChannel.⌋

[constr_3004] Clustering and separation shall be exclusive

Imposition time: CP: IT_SysDesc

⌈Clustering and separation shall be exclusive, i.e. it SHALL NOT be possible that two SwComponentPrototypes A and B are associated both by a ComponentClustering and by a ComponentSeparation at the same time.⌋

[constr_3005] valid EcuResourceEstimation

Imposition time: CP: IT_SysDesc

⌈The same EcuInstance shall be referenced directly from the EcuResourceEstimation and from the SwcToEcuMapping:

EcuResourceEstimation.swCompToEcuMapping.ecuInstance == EcuResourceEstimation.ecuInstance⌋

[constr_3006] valid EcuMapping

Imposition time: CP: IT_SysDesc

⌈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⌋

[constr_3007] selectorFieldCodes for dynamic part alternatives

Imposition time: CP: IT_SysDesc

⌈The selectorFieldCodes for the dynamic part alternatives within one MultiplexedIPdu shall differ from each other.⌋

[constr_3008] EcuInstance subelements

Imposition time: CP: IT_SysDesc

⌈The CommunicationConnector and the CommunicationController that is referenced by the CommunicationConnector shall be owned by the same EcuInstance.⌋

[constr_3009] Overlapping of `ISignal`s is prohibited

Imposition time: CP: `IT_SysDesc`

⌈`ISignal`s mapped to an `ISignalIPdu` shall not overlap.⌋

[constr_3010] `ISignalIPdu` length shall not be exceeded

Imposition time: CP: `IT_SysDesc`

⌈The combined length of all `ISignal`s and `updateIndicationBitPosition`s that are mapped into an `ISignalIPdu` shall not exceed the defined `Pdu length`.⌋

[constr_3011] Overlapping of `updateIndicationBits` of `ISignal`s is prohibited

Imposition time: CP: `IT_SysDesc`

⌈The `updateIndicationBitPosition` for an `ISignal` in an `ISignalIPdu` shall not overlap with other `updateIndicationBitPosition`s or `ISignal` locations.⌋

[constr_3012] Overlapping of `Pdu`s is prohibited

Imposition time: CP: `IT_SysDesc`

⌈`Pdu`s mapped to a `FlexrayFrame` shall NOT overlap.⌋

[constr_3013] `FlexrayFrame` length shall not be exceeded

Imposition time: CP: `IT_SysDesc`

⌈The combined length of all `Pdu`s that are mapped into a `FlexrayFrame` shall not exceed the defined `FlexrayFrame` length.⌋

[constr_3014] Overlapping of updateIndicationBits for `Pdu`s is prohibited

Imposition time: CP: `IT_SysDesc`

⌈The `updateIndicationBitPosition` for a `Pdu` in a `FlexrayFrame` shall NOT overlap with other `updateIndicationBitPosition`s and `Pdu` locations.⌋

[constr_3015] Number of LIN channels

Imposition time: CP: `IT_SysDesc`

⌈LIN clusters shall aggregate exactly one `LinPhysicalChannel`.⌋

[constr_3018] Number of FlexRay channels

Imposition time: CP: `IT_SysDesc`

⌈A `FlexrayCluster` shall use either one `FlexrayPhysicalChannel` with `channelName` set to either `channelA` or `channelB` or else two `FlexrayPhysicalChannel`s with one `channelName channelA` and one `channelName channelB`.⌋

[constr_3019] In the flat ECU extract each required interface shall be satisfied by connected provided interfaces

Imposition time: CP: IT_EcuExt

⌈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.⌋

[constr_3020] communicationDirection of containedISignalIPduGroups

Imposition time: CP: IT_SysDesc

⌈The value of the attribute communicationDirection of containedISignalIPduGroup shall be identical to the value of the attribute communicationDirection of the enclosing ISignalIPduGroup.⌋

[constr_3025] Usage of NPdus in TpConnections

Imposition time: CP: IT_SysDesc

⌈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, and J1939Tp).⌋

[constr_3027] Existence of ecuExtractVersion

Imposition time: CP: IT_EcuExt

⌈In case the category of the System is SYSTEM_EXTRACT or ECU_EXTRACT the ecuExtractVersion attribute shall be defined.⌋

[constr_3028] FibexElements

Imposition time: CP: IT_SysDesc

⌈Each FibexElement that is used in the System Description shall be referenced by the System element in the role FibexElement.⌋

[constr_3029] Assign-Frame command usage

Imposition time: CP: IT_SysDesc

⌈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_3030] valid relationship between ECUMapping and EcuInstance

Imposition time: CP: IT_SysDesc

⌈If an EcuInstance is assigned to a HwElement the EcuInstance shall belong to the same System as the ECUMapping.⌋

[constr_3031] Complete System Description does not have ports on the outermost composition

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3036] Pdus in CAN and LIN Frames

Imposition time: CP: IT_SysDesc

⌈CAN Frames and LIN Frames shall only contain one Pdu.⌋

[constr_3037] maximum Frame frameLength for CAN and LIN

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈For FlexRay the maximum frameLength is 254 bytes.⌋

[constr_3039] pncIdentifier range

Imposition time: CP: IT_SysDesc

⌈The pncIdentifier value shall be in the range of 8..63 for normal CAN and in the range of 8..511 for CAN FD, FlexRay and Ethernet.⌋

[constr_3040] Restriction of pncIdentifier values

Imposition time: CP: IT_SysDesc

⌈The pncIdentifier value shall be within the range described by pncVectorOffset and pncVectorLength.⌋

[constr_3044] CBV configuration in case partial network is used

Imposition time: CP: IT_SysDesc

⌈In case a partial network is used the control bit vector (CBV) shall be defined in Byte 0 of the NmPdu (nmCbvPosition = 0).⌋

[constr_3045] Signal content evaluation vs. Mode evaluation

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3046] Consistency of `TransmissionModeCondition.iSignalInIPdu`

Imposition time: CP: `IT_SysDesc`

⌈The `ISignalToIPduMapping` referenced by the `TransmissionModeCondition` in the role `iSignalInIPdu` shall belong to the same `ISignalIPdu` as the `TransmissionModeCondition`.⌋

[constr_3047] Uniqueness of `macMulticastAddress`es

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `macMulticastAddress` shall be unique in a particular `EthernetCluster`.⌋

[constr_3048] Range of `vlanIdentifier`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The allowed values of `vlanIdentifier` range from 0 to 4095.⌋

[constr_3050] `J1939Cluster` uses exactly one `CanPhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈A `J1939Cluster` shall aggregate exactly one `CanPhysicalChannel`.⌋

[constr_3051] Restriction of `ISignalMapping` references

Imposition time: CP: `IT_SysDesc`

⌈If the `sourceSignal` references an `ISignal` then the `targetSignal` shall also reference an `ISignal`.⌋

[constr_3052] Complete `ISignalMapping` of `ISignalGroup` signals

Imposition time: CP: `IT_SysDesc`

⌈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.⌋

[constr_3053] Complete `ISignalMapping` of target `ISignalGroup`

Imposition time: CP: `IT_SysDesc`

⌈If an `ISignalGroup` is referenced by a `targetSignal` then [TPS_SYST_02162] applies for each of the contained `ISignal` of that `ISignalGroup`.⌋

[constr_3057] Maximal one `BusspecificNmEcu` per `NmEcu` and bus system is allowed to be defined

Imposition time: CP: `IT_SysDesc`

⌈For each `NmEcu`, at most one `BusspecificNmEcu` per bus system (FlexRay/Can/Udp/J1939) is allowed to be defined.⌋

[constr_3060] Allowed Attributes for networkRepresentationProps and physicalProps

Imposition time: CP: IT_SysDesc

⌈

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	D	NA
swValueBlockSize	NA	NA
unit	D	M
valueAxisDataType	NA	NA

⌋

[constr_3062] The EcuInstance that is referenced from a specific CouplingElement shall be connected to the same EthernetCluster as the specific CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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.⌋

[constr_3067] initValue defined in the context of ISignal

Imposition time: CP: IT_SysDesc

⌈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_3068] DoIpPowerModeStatusNeeds in the category ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈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_3069] Allowed CanNmCluster.nmNidPosition values

Imposition time: CP: IT_SysDesc

⌈If defined, the value of CanNmCluster.nmNidPosition shall only be set to either 0 or 1.⌋

[constr_3070] Allowed CanNmCluster.nmCbvPosition values

Imposition time: CP: IT_SysDesc

⌈If defined, 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

Imposition time: CP: IT_SysDesc

⌈CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value.⌋

[constr_3073] nmVoteInformation only valid for FrNm

Imposition time: CP: IT_SysDesc

⌈The nmVoteInformation attribute is only valid for FrNm.⌋

[constr_3074] No TransmissionAcknowledgementRequest for multiple senders

Imposition time: CP: IT_EcuExt

⌈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.⌋

[constr_3078] Allowed `UdpNmCluster.nmNidPosition` values

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `UdpNmCluster.nmNidPosition` shall only be set to either 0 or 1.⌋

[constr_3079] Allowed `UdpNmCluster.nmCbvPosition` values

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, 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

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈`UdpNmCluster.nmCbvPosition` and `UdpNmCluster.nmNidPosition` shall never have the same value.⌋

[constr_3081] Value of category in `GeneralPurposePdu`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `category` of `GeneralPurposePdu` can have the following values:

- SD (Service Discovery)
- GLOBAL_TIME
- DoIP

⌋

[constr_3082] Value of category in `GeneralPurposeIPdu`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `category` of `GeneralPurposeIPdu` can have the following values:

- XCP
- SOMEIP_SEGMENTED_IPDU
- DLT
- IDS
- VDP

⌋

[constr_3083] Exactly one AtomicSwComponentType on an EcuInstance may use GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange

Imposition time: CP: IT_EcuExt

⌈The Dem only supports exactly one AtomicSwComponentType using GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange on one EcuInstance.⌋

[constr_3084] Service port in the role PowerTakeOff

Imposition time: CP: IT_EcuExt

⌈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

Imposition time: CP: IT_EcuExt

⌈Within the context of one EcuInstance, there can only be one service port that uses the role CallbackDCMRequestServices in the RoleBasedPortAssignment.role.⌋

[constr_3086] Role of SystemSignal in n:1 sender-receiver communication

Imposition time: CP: IT_EcuExt

⌈In case of n:1 communications

- if DataTransformation is used each sender shall be mapped to the same SystemSignal
- if DataTransformation is not used each sender shall be mapped
  - to the same SystemSignal in case of a primitive DataType on the sender side,
  - to the same SystemSignalGroup in case of a composite DataType on the sender side.

⌋

[constr_3090] TpSdu transmission on a PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3094] Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3095] canControllerFdAttributes and canControllerFdRequirements are mutually exclusive

Imposition time: CP: IT_SysDesc

⌈The existence of canControllerFdAttributes and canControllerFd-Requirements is mutually exclusive.⌋

[constr_3096] Allowed values for diagnosticMessageType

Imposition time: CP: IT_SysDesc

⌈The allowed values of diagnosticMessageType range from 1..60.⌋

[constr_3097] Overlapping of segments of one MultiplexedIPdu is not allowed

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈The segments defined by the SegmentPosition elements aggregated in the DynamicPart of a MultiplexedIPdu shall not exceed the length of any Dynamic-PartAlternative.iPdu.⌋

[constr_3100] Defined segments in a StaticPart shall not exceed the length of the StaticPart.iPdu

Imposition time: CP: IT_SysDesc

⌈The segments defined by the SegmentPosition elements aggregated in the StaticPart of a MultiplexedIPdu shall not exceed the length of the Static-Part.iPdu⌋

[constr_3101] Signal representation of selector field for `DynamicPartAlternative`

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_3102] Restriction on usage of `J1939NodeName` attributes

Imposition time: CP: `IT_SysDesc`

⌈`EcuInstance`s that are referenced by the same `CanPhysicalChannel` of a `J1939Cluster` via `commConnector` shall not aggregate two `J1939Node`s with identical `J1939NodeName` attributes.⌋

[constr_3103] Range of `ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `ecuInstance` range from 0 to 7.⌋

[constr_3104] Range of `function`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `function` range from 0 to 255.⌋

[constr_3105] Range of `functionInstance`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `functionInstance` range from 0 to 31.⌋

[constr_3106] Range of `identitiyNumber`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `identitiyNumber` range from 0 to 2097151.⌋

[constr_3107] Range of `industryGroup`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `industryGroup` range from 0 to 7.⌋

[constr_3108] Range of `manufacturerCode`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `manufacturerCode` range from 0 to 2047.⌋

[constr_3109] Range of `vehicleSystem`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `vehicleSystem` range from 0 to 127.⌋

[constr_3110] Range of vehicleSystemInstance

Imposition time: CP: IT_SysDesc

⌈The allowed values of vehicleSystemInstance range from 0 to 15.⌋

[constr_3111] returnSignal in ClientServerToSignalMapping is mandatory

Imposition time: CP: IT_EcuExt

⌈A ClientServerToSignalMapping shall always have a returnSignal defined.⌋

[constr_3112] Invalidation support for partial mapping of a data element typed by composite data type

Imposition time: CP: IT_EcuExt

⌈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_3113] AbstractEthernetFrame shall not have a PduToFrameMapping

Imposition time: CP: IT_SysDesc

⌈It is not allowed to map Pdus into AbstractEthernetFrames.⌋

[constr_3114] FlatInstanceDescriptors pointing to the same ParameterDataPrototype shall have different postBuildVariantConditions

Imposition time: CP: IT_EcuExt

⌈FlatInstanceDescriptors that are pointing as an atpTarget to the same ParameterDataPrototype instance shall have different postBuildVariantConditions.⌋

[constr_3115] FlatInstanceDescriptors pointing to the same ParameterDataPrototype instance

Imposition time: CP: IT_EcuExt

⌈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.⌋

[constr_3116] Overlap of ClientIdRanges in the context of the enclosing System

Imposition time: CP: IT_SysDesc

⌈The ClientIdRange defined for an EcuInstance shall not overlap with the ClientIdRange of any other EcuInstance or MachineDesigns in the context of the enclosing System.⌋

[constr_3117] Allowed value of attribute clientId

Imposition time: CP: IT_SysDesc

⌈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 if the EcuInstance aggregates a ClientIdRange.⌋

[constr_3118] Valid reference target for ClientIdDefinition.clientServerOperation.contextPort

Imposition time: CP: IT_SysDesc

⌈In the context of the definition of a ClientIdDefinition, the reference clientServerOperation.contextPort shall only refer to an RPortPrototype.⌋

[constr_3121] The length of transformer chains is limited to 255 transformers

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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 TransformationTechnologys referenced by the same DataTransformation.⌋

[constr_3123] Serializer transformer shall be the first in a chain

Imposition time: CP: IT_SysDesc

⌈A serializer transformer (TransformationTechnology with attribute transformerClass set to serializer) shall be the first transformer in a transformer chain.⌋

[constr_3124] Applicability of needsOriginalData

Imposition time: CP: IT_SysDesc

⌈The attribute needsOriginalData of a TransformationTechnology shall only be used for the non-first transformers in the transformer chain.⌋

[constr_3125] Value of attribute inPlace for the first transformer in a chain

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3127] Certain ISignals always need a reference to DataTransformation

Imposition time: CP: IT_SysDesc

⌈An ISignal which references a SystemSignal which is referenced by a SystemSignalGroup in the role transformingSystemSignal shall always reference a DataTransformation.⌋

[constr_3128] SOME/IP transformer configuration

Imposition time: CP: IT_SysDesc

⌈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).

⌋

[constr_3129] Byte Order of SOME/IP transformer

Imposition time: CP: IT_SysDesc

⌈The attribute byteOrder of SOMEIPTransformationDescription shall be different from opaque.⌋

[constr_3130] Range of Interface Version

Imposition time: CP: IT_SysDesc

⌈The value of the attribute interfaceVersion shall be in the range $[0; 255]$⌋

[constr_3132] Required COM Based Transformation for comBasedSignalGroupTransformation

Imposition time: CP: IT_SysDesc

⌈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 [8].⌋

[constr_3133] physicalLayerType of connected CouplingPorts

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The connectionNegotiationBehavior of two CouplingPorts which are connected via a CouplingPortConnection shall not be both set to slave.⌋

[constr_3136] Allowed payload of SecuredIPdus

Imposition time: CP: IT_SysDesc

⌈SecuredIPdus are allowed to reference PduTriggerings of ISignalIPdus, ContainerIPdus, DcmIPdus, MultiplexedIPdus, GeneralPurposeIPdus with category SOMEIP_SEGMENTED_IPDU or VDP, and UserDefinedIPdus.⌋

[constr_3137] IPduPort.rxSecurityVerification is configurable on the receiver side

Imposition time: CP: IT_SysDesc

⌈The IPduPort.rxSecurityVerification attribute shall only be used in IPduPorts with the communicationDirection = in.⌋

[constr_3138] IPduPort.rxSecurityVerification validness

Imposition time: CP: IT_SysDesc

⌈The IPduPort.rxSecurityVerification information is only valid for SecuredIPdus.⌋

[constr_3140] No ByteOrderEnum.opaque allowed for System.containerIPduHeaderByteOrder

Imposition time: CP: IT_SysDesc

⌈The values of System.containerIPduHeaderByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

[constr_3141] Only IPdus shall be part of a ContainerIPdu

Imposition time: CP: IT_SysDesc

⌈The PduTriggering which is referenced in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering shall refer to a subclass of an IPdu in the role PduTriggering.iPdu.⌋

[constr_3142] Mandatory headerIdLongHeader for longHeader

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = longHeader the ContainedIPduProps.headerIdLongHeader shall be defined.⌋

[constr_3143] Mandatory headerIdShortHeader for shortHeader

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = shortHeader the ContainedIPduProps.headerIdShortHeader shall be defined.⌋

[constr_3146] Partial Networking timing constraint

Imposition time: CP: IT_SysDesc

⌈For Partial Networking the following timing constraints shall be ensured:

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

⌋

[constr_3148] executeDespiteDataUnavailability setting in case an E2E Transformer is used

Imposition time: CP: IT_SysDesc

⌈A transformer chain using E2E shall be configured with DataTransformation.executeDespiteDataUnavailability = TRUE.⌋

[constr_3149] `TransformationTechnology.needsOriginalData` settings for E2E Transformer

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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_08: `BufferProperties.headerLength` = 128 bits
8. PROFILE_11: `BufferProperties.headerLength` = 16 bits
9. PROFILE_22: `BufferProperties.headerLength` = 16 bits
10. PROFILE_04m: `BufferProperties.headerLength` = 128 bits
11. PROFILE_07m: `BufferProperties.headerLength` = 192 bits
12. PROFILE_08m: `BufferProperties.headerLength` = 160 bits
13. PROFILE_44: `BufferProperties.headerLength` = 96 bits
14. PROFILE_44m: `BufferProperties.headerLength` = 128 bits

⌋

[constr_3152] `BufferProperties.headerLength` settings for any transformer used in combination with a COM Based transformer

Imposition time: CP: `IT_SysDesc`

⌈A transformer used in a transformer chain with a COM Based transformer shall be configured with the following values:

- `BufferProperties.headerLength` = 0

⌋

[constr_3153] E2E header field reservation required by COM Based transformer

Imposition time: CP: IT_SysDesc

⌈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_01: if dataIdMode == lower12Bit: 16 bits

PROFILE_01: if dataIdMode != lower12Bit: 12 bits

PROFILE_02: 16 bits

PROFILE_04: 96 bits

PROFILE_05: 24 bits

PROFILE_06: 40 bits

PROFILE_07: 160 bits

PROFILE_08: 128 bits

PROFILE_11: if dataIdMode == lower12Bit: 16 bits

PROFILE_11: if dataIdMode == all16Bit: 12 bits

PROFILE_22: 16 bits

PROFILE_04m: 128 bits

PROFILE_07m: 192 bits

PROFILE_08m: 160 bits

PROFILE_44: 96 bits

PROFILE_44m: 128 bits

PROFILE_76: 40 bits

⌋

[constr_3155] Allowed values for EndToEndTransformationDescription.upperHeaderBitsToShift

Imposition time: CP: IT_SysDesc

⌈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_3156] Allowed values for `EndToEndTransformationISignalProps.dataId` in PROFILE_01 and PROFILE_11

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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_3159] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_04, PROFILE_04m PROFILE_44 and PROFILE_44m

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_04m, PROFILE_44, or PROFILE_44m the value of `maxDeltaCounter` attribute shall be in the range 1-65535.⌋

[constr_3160] `EndToEndTransformationISignalProps.dataId` in PROFILE_02 and PROFILE_22

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 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, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the attribute EndToEndTransformationISignalProps.dataLength shall be 0.⌋

[constr_3165] Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01, PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_3166] EndToEndTransformationDescription.upperHeaderBitsToShift in PROFILE_02

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02 then the value of the upperHeaderBitsToShift attribute shall be 0.⌋

[constr_3167] Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m the value of the EndToEndTransformationDescription.offset attribute shall be equal to the value of the EndToEndTransformationDescription.upperHeaderBitsToShift attribute.⌋

[constr_3169] EndToEndTransformationDescription.offset value in PROFILE_02, PROFILE_22 and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02, PROFILE_22, or PROFILE_76 then the value of the EndToEndTransformationDescription.offset attribute shall be 0.⌋

[constr_3172] Effect of `EndToEndTransformationDescription.profileBehavior` value in PROFILE_01

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 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.

⌋

[constr_3182] Restriction on TransformationTechnology.transformationDescription VariationPoint

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3183] ISignalGroup with transformationISignalProps

Imposition time: CP: IT_SysDesc

⌈An ISignalGroup that aggregates transformationISignalProps shall reference the DataTransformation in the role comBasedSignalGroupTransformation.⌋

[constr_3184] Only one EndToEndTransformationISignalProps.dataId element in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_3185] Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.dataIdMode attribute shall be 0.⌋

[constr_3187] Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.counterOffset attribute shall be 0.⌋

[constr_3189] Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.crcOffset attribute shall be 0.⌋

[constr_3191] Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_01, PROFILE_11 and dataIdMode equal to lower12Bit

Imposition time: CP: IT_SysDesc

⌈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_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76 or dataIdMode different from lower12Bit

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 or the EndToEndTransformationDescription.dataIdMode attribute is set to value different from lower12Bit then the multiplicity of the EndToEndTransformationDescription.dataIdNibbleOffset attribute shall be 0.⌋

[constr_3193] Multiplicity of EndToEndTransformationDescription.offset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.offset attribute shall be 1.⌋

[constr_3195] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02 and PROFILE_22

Imposition time: CP: IT_SysDesc

⌈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_3196] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_05

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_06 then the attribute maxDeltaCounter shall be in the range 1-255.⌋

[constr_3198] Uniqueness of PncMapping.shortLabel

Imposition time: CP: IT_SysDesc

⌈If the optional shortLabel attribute is used it shall be unique in the System scope.⌋

[constr_3199] ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation

Imposition time: CP: IT_SysDesc

⌈In a complete model every ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation.⌋

[constr_3202] LinFrameTriggering to LinUnconditionalFrame reference restriction in LinEventTriggeredFrame context

Imposition time: CP: IT_SysDesc

⌈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 LinEventTriggered-Frame in the role linUnconditionalFrame.⌋

[constr_3203] LinFrameTriggering to LinSporadicFrame reference restriction in LinSporadicFrame context

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3204] `LinUnconditionalFrame`s associated with a `LinSporadicFrame`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈A `FrameTriggering` that references a `LinSporadicFrame` shall not have a reference to a `FramePort`.⌋

[constr_3206] Existence of `FramePort` for a `FrameTriggering` that references a `LinEventTriggeredFrame`

Imposition time: CP: `IT_SysDesc`

⌈A `FrameTriggering` that references a `LinEventTriggeredFrame` shall not have a reference to a `FramePort`.⌋

[constr_3208] `executeDespiteDataUnavailability` usage restriction for received `ISignal`s

Imposition time: CP: `IT_SysDesc`

⌈For each `SystemSignal`, the collection of `ISignal`s that refer to the `SystemSignal` and also refer to an `ISignalPort` on the same `EcuInstance` with `communicationDirection` = `in` shall not include an `ISignal` that references a `DataTransformation` where `executeDespiteDataUnavailability` is set to true.⌋

[constr_3209] `CanFrameTriggering`s with identical PGN

Imposition time: CP: `IT_SysDesc`

⌈For all `CanFrameTriggering`s where the attribute `identifier` contains the identical PGN (as defined in section 5.2 Protocol Data Unit in [9]) the attribute `j1939requestable` shall also have an identical value.⌋

[constr_3210] `J1939TpPg`s with identical `pgn` value

Imposition time: CP: `IT_SysDesc`

⌈For all `J1939TpPg`s where the attribute `pgn` has an identical value the attribute `requestable` shall also have an identical value.⌋

[constr_3211] `PduTriggering`s with `triggerIPduSendCondition`

Imposition time: CP: `IT_SysDesc`

⌈Only `PduTriggering`s with references to `ISignalIPdu`s are allowed to contain a `triggerIPduSendCondition`.⌋

[constr_3212] Limitation of DoIpTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈DoIpTpConnection shall only reference PduTriggerings of DcmIPdus or UserDefinedIPdus in the role tpSdu.⌋

[constr_3213] TransformationISignalProps.csErrorReaction setting in case that the serializer transformerClass and Client/Server communication is used

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3215] TransformationTechnology.version and TransformationTechnology.protocol settings for request and response of a client/server communication on the same TransformationTechnology

Imposition time: CP: IT_SysDesc

⌈TransformationTechnology.version and TransformationTechnology.protocol shall be identical for ISignals that are derived from the same ClientServerOperation and TransformationTechnology. This means that all ISignals that refer to ClientServerToSignalMapping.callSignal or to ClientServerToSignalMapping.returnSignal of the same ClientServerToSignalMapping and TransformationTechnology shall have the same TransformationTechnology.protocol and TransformationTechnology.version defined.⌋

[constr_3218] Range of Size of Array Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfArrayLengthFields of SOMEIPTransformationISignalProps shall be either 0, 1, 2 or 4.⌋

[constr_3219] The mutual existence of LinSlaves in the LinMaster EcuExtract

Imposition time: CP: IT_EcuExt

⌈LinSlaves shall not be part of the EcuExtract of the corresponding LinMaster.⌋

[constr_3220] Range of Size of Structure Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfStructLengthFields of SOMEIPTransformationISignalProps shall be either 0, 1, 2 or 4.⌋

[constr_3221] Range of Size of Union Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfUnionLengthFields of SOMEIPTransformation-ISignalProps shall be either 0, 1, 2 or 4.⌋

[constr_3222] No ByteOrderEnum.opaque allowed for PduToFrame-Mapping.packingByteOrder

Imposition time: CP: IT_SysDesc

⌈The values of PduToFrameMapping.packingByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.most-SignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

[constr_3223] No ByteOrderEnum.opaque allowed for MultiplexedIPdu.selectorFieldByteOrder

Imposition time: CP: IT_SysDesc

⌈The values of MultiplexedIPdu.selectorFieldByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.most-SignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

[constr_3224] No ByteOrderEnum.opaque allowed for SegmentPosition.segmentByteOrder.

Imposition time: CP: IT_SysDesc

⌈The values of SegmentPosition.segmentByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.most-SignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

[constr_3225] LinFrameTriggering.linChecksum not allowed for LinSporadicFrames

Imposition time: CP: IT_SysDesc

⌈The linChecksum attribute of a LinFrameTriggering that references a LinSporadicFrame shall not be set.⌋

[constr_3226] `LinFrameTriggering.linChecksum` for `LinEventTriggeredFrame`s

Imposition time: CP: `IT_SysDesc`

⌈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_3227] `NmNode.nmPassiveModeEnabled` setting

Imposition time: CP: `IT_SysDesc`

⌈`NmNode.nmPassiveModeEnabled` shall be set to the same value in all `NmCluster`s with the same bus protocol in the scope of one `NmEcu`.⌋

[constr_3229] `SwComponentPrototype` mapped to an `ApplicationPartition` and `EcuInstance`

Imposition time: CP: `IT_EcuExt`

⌈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`.⌋

[constr_3230] Usage of `SenderRecRecordElementMapping.applicationRecordElement`

Imposition time: CP: `IT_EcuExt`

⌈`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_3231] Usage of `IndexedArrayElement.applicationArrayElement`

Imposition time: CP: `IT_EcuExt`

⌈`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_3232] `ApplicationPartition` is allowed to be mapped to only one `EcuPartition`

Imposition time: CP: `IT_EcuExt`

⌈Each `ApplicationPartition` shall be mapped at most once to an `EcuPartition` via the `ApplicationPartitionToEcuPartitionMapping`.⌋

[constr_3243] `FrameTriggering.pduTriggering` condition

Imposition time: CP: `IT_SysDesc`

⌈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_3244] Usage of `SenderRecRecordElementMapping.implementationRecordElement`

Imposition time: CP: `IT_EcuExt`

⌈`SenderRecRecordElementMapping.implementationRecordElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping.dataElement`) is typed by an `ImplementationDataType`.⌋

[constr_3245] Usage of `IndexedArrayElement.implementationArrayElement`

Imposition time: CP: `IT_EcuExt`

⌈`IndexedArrayElement.implementationArrayElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping.dataElement`) is typed by an `ImplementationDataType`.⌋

[constr_3246] `Frame.packingByteOrder` mix within a `Frame` is not allowed

Imposition time: CP: `IT_SysDesc`

⌈All `PduToFrameMapping`s within a `Frame` shall have the same `packingByteOrder` value.⌋

[constr_3247] Byte order mix within a `MultiplexedIPdu` is not allowed

Imposition time: CP: `IT_SysDesc`

⌈The `segmentByteOrder` of all `SegmentPosition`s and the `selectorFieldByteOrder` shall have the same value in the `MultiplexedIPdu`.⌋

[constr_3248] Category of `HwElement` for `ECUMapping`

Imposition time: CP: `IT_SysDesc`

⌈The `HwElement` which is referenced from `ECUMapping` in the role `ecu` shall be of category `MicroController`⌋

[constr_3250] `PduTriggering.iSignalTriggering` condition

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_3252] `ISignalTriggering.iSignalPort` reference condition

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈A `PduTriggering` shall only reference an `ISignalTriggering` in the role `iSignalTriggering` if both the `PduTriggering` and `ISignalTriggering` are aggregated by the same `PhysicalChannel`.⌋

[constr_3258] Restriction on `ISignal.length` in case `iSignalType` is set to `array`

Imposition time: CP: `IT_SysDesc`

⌈If `ISignal.iSignalType` is set to `array` then `ISignal.length` shall be a multiple of 8.⌋

[constr_3261] `GlobalTimeDomain.pduTriggering` category

Imposition time: CP: `IT_SysDesc`

⌈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.⌋

[constr_3262] ConsumedEventGroup.eventGroupIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedEventGroup.eventGroupIdentifier is mandatory.⌋

[constr_3263] Restriction of usage of SwcToEcuMapping in a System

Imposition time: CP: IT_SysDesc

⌈For all SwcToEcuMappings in a System the following restriction applies: No two SwcToEcuMappings shall have the exact same reference to

- SwComponentPrototype
- EcuInstance

⌋

[constr_3264] Server side ClientServerToSignalMappings in case of a n:1 inter-ECU client-server communication

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3265] TransformationTechnology.hasInternalState setting for an E2E transformer

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈The value of hasInternalState shall be set to true for a SOME/IP Transformer if the ISignal that is referencing this transformer is mapped into an ISignalIPdu and this ISignalIPdu is referenced by a PduTriggering that in turn is referenced by a SomeipTpConnection in the role tpSdu or tpConcurrentProcessingSdu.⌋

[constr_3267] PduTriggerings in Service Discovery StaticSocketConnections

Imposition time: CP: IT_SysDesc

⌈SD StaticSocketConnections defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall only refer to PduTriggerings which point to GeneralPurposePdus of category SD.⌋

[constr_3268] Service Discovery `StaticSocketConnection` aggregation by a `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall be aggregated by a `SocketAddress` that in turn aggregates an `ApplicationEndpoint` that defines a Udp Port.⌋

[constr_3269] Service Discovery `StaticSocketConnection.remoteAddress` reference to a `TpPort`

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` with Udp Port `portNumber` set to 0. This means that any remote port number is accepted for receiving and for sending, i.e., that the remote port number is configured at runtime.⌋

[constr_3270] Service Discovery `StaticSocketConnection.remoteAddress` reference to an IP Address

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Address ANY (IPv4 or IPv6). This means that any remote IP address is accepted for receiving and for sending, i.e., that the remote IP address is configured at runtime.⌋

[constr_3272] `SoConIPduIdentifier.headerId` setting for SD `StaticSocketConnection`s

Imposition time: CP: `IT_SysDesc`

⌈The `SoConIPduIdentifier.headerId` of SD `StaticSocketConnection`s defined in [TPS_SYST_02414] and [TPS_SYST_02415] shall always be set to 0x FFFF8100 for SD messages.⌋

[constr_3273] Service Discovery multicast `StaticSocketConnection`'s aggregation by an `ApplicationEndpoint`

Imposition time: CP: `IT_SysDesc`

⌈The SD `StaticSocketConnection` for multicast defined in [TPS_SYST_02415] 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`

Imposition time: CP: `IT_SysDesc`

⌈The SD `StaticSocketConnection` for unicast defined in [TPS_SYST_02414] shall be aggregated by an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Unicast Address.⌋

[constr_3276] Prohibition of usage of `allowedIPv6ExtHeaders` in IPv4 `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈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 `IPv6ExtHeaderFilterList`s in IPv6 `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈All `SocketAddress`es related to the same IPv6 `NetworkEndpoint` shall all reference either no or exactly the same `IPv6ExtHeaderFilterList` with the `allowedIPv6ExtHeaders` attribute.⌋

[constr_3278] Usage of `SOMEIPTransformationProps.sizeOfArrayLengthField`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: IT_SysDesc

⌈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 arrays in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈The definition of DataPrototypeTransformationProps.transformationProps.alignment is only allowed if the SOMEIPTransformationDescription.alignment is not defined.⌋

[constr_3297] Prohibition of usage of allowedTcpOptions in Udp SocketAddress

Imposition time: CP: IT_SysDesc

⌈Udp SocketAddress shall not define allowedTcpOptions. A Udp SocketAddress aggregates an ApplicationEndpoint that has a UdpTp defined as tpConfiguration.⌋

[constr_3298] `Ipv6Configuration.ipv6Address` range in case of `enableAnycast`

Imposition time: CP: `IT_SysDesc`

⌈If `Ipv6Configuration.enableAnycast` is set to true then the `Ipv6Configuration.ipv6Address` needs to be in the unicast addressing range.⌋

[constr_3311] Usage of `SocketAddress.flowLabel`

Imposition time: CP: `IT_SysDesc`

⌈`SocketAddress.flowLabel` shall only be used if the aggregated `ApplicationEndpoint` refers to a `NetworkEndpoint` with an `Ipv6Configuration`.⌋

[constr_3312] Consistency of `vlanPriority` and `EthernetCommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈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.⌋

[constr_3313] E2E transformer configuration

Imposition time: CP: `IT_SysDesc`

⌈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`

⌋

[constr_3316] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_07, PROFILE_08, PROFILE_07m and PROFILE_08m

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_07, PROFILE_08, PROFILE_07m, or PROFILE_08m the value of `maxDeltaCounter` attribute shall be in the range 1-4’294’967’295.⌋

[constr_3317] Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the ImplementationDataTypes

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3318] Allowed use of ISignal.networkRepresentationProps

Imposition time: CP: IT_SysDesc

⌈If a reference from ISignal to DataTransformation in the role dataTransformation exists, this ISignal SHALL NOT aggregate SwDataDefProps in the role networkRepresentationProps.⌋

[constr_3319] Existence of DataPrototypeTransformationProps.networkRepresentationProps

Imposition time: CP: IT_SysDesc

⌈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

⌋

[constr_3322] Consistent setting of SoConIPduIdentifier.pduCollectionSemantics in the context of one SocketAddress

Imposition time: CP: IT_SysDesc

⌈The value of the attribute SoConIPduIdentifier.pduCollectionSemantics shall be identical for all referenced SoConIPduIdentifiers within the context of a given SocketAddress.⌋

[constr_3323] Relation between NmCluster.nmPncParticipation and PncMapping.pncGroup

Imposition time: CP: IT_SysDesc

⌈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_3324] Category of SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3325] SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute value settings for standardized AUTOSAR security profiles

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈

Attributes	PROFILE_01	PROFILE_02	PROFILE_03	PROFILE_04
algorithmFamily	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES
algorithmMode	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ GCM_ SIV
length	128 bits	128 bits	128 bits	128 bits
authInfoTxLength	24 bits	24 bits	28 bits	128 bits
freshnessValueLength	Not specified	0 bits	64 bits	96 bits
freshnessValueTx-Length	8 bits	0 bits	4 bits	96 bits

⌋

[constr_3326] Allowed values for EndToEndTransformationDescription.dataIdMode in PROFILE_11

Imposition time: CP: IT_SysDesc

⌈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_3328] SomeipTpConnection.transportPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈A PduTriggering that is referenced by a SomeipTpConnection in the role tpSdu shall reference an IPdu in the role iPdu.⌋

[constr_3330] Same transportPdu shall not be used in different SomeipTpConnections

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3331] Standardized values for the attribute category of meta-class EthernetCommunicationConnector

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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
- CAN_XL: This represents the tunneling of Ethernet frames handled by the EthernetCommunicationConnector through CAN XL.

⌋

[constr_3332] Standardized values for the attribute `category` of meta-class `EthernetCommunicationController`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈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
- CAN_XL: This represents the tunneling of Ethernet frames handled by the `EthernetCommunicationController` through CAN XL.

⌋

[constr_3333] Standardized values for the attribute `category` of meta-class `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈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

⌋

[constr_3334] Allowed references between `EthernetPhysicalChannel` and `EthernetCommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈An `EthernetPhysicalChannel` is only allowed to reference `EthernetCommunicationConnector`s in the role `commConnector` that have the same `category` value as the referencing `EthernetPhysicalChannel`.⌋

[constr_3335] Allowed references between `EthernetCommunicationConnector` and `EthernetCommunicationController`

Imposition time: CP: `IT_SysDesc`

⌈An `EthernetCommunicationConnector` is only allowed to reference an `EthernetCommunicationController` in the role `commController` that has the same `category` value as the referencing `EthernetCommunicationConnector`.⌋

[constr_3336] `EthernetPhysicalChannel.soAdConfig` in case of WIRELESS `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈If `EthernetPhysicalChannel` has the `category` *WIRELESS* then the `EthernetPhysicalChannel` shall not aggregate the `SoAdConfig`.⌋

[constr_3337] `IPduPort.useAuthDataFreshness` is configurable on the receiver side

Imposition time: CP: `IT_SysDesc`

⌈The `IPduPort.useAuthDataFreshness` attribute shall only be used in `IPduPort`s with the `communicationDirection` = in.⌋

[constr_3338] `IPduPort.useAuthDataFreshness` validness

Imposition time: CP: `IT_SysDesc`

⌈The `IPduPort.useAuthDataFreshness` information is only valid for `SecuredIPdu`s.⌋

[constr_3339] Relation between `authDataFreshnessStartPosition`, `authDataFreshnessLength` and `useAuthDataFreshness`

Imposition time: CP: `IT_SysDesc`

⌈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 `IPduPort`s with `communicationDirection` = in that are referenced by a `PduTriggering` of the `SecuredIPdu`.⌋

[constr_3364] `headerLength` shall be a multiple of 8

Imposition time: CP: `IT_SysDesc`

⌈The header length in bits specified by `headerLength` shall be a multiple of 8.⌋

[constr_3365] `EthernetPhysicalChannel`s with different `category` values are not allowed within an `EthernetCluster`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A mix of `EthernetPhysicalChannel`s with different `category` values within an `EthernetCluster` is currently not supported by AUTOSAR.⌋

[constr_3373] Limitation on the number of `PhysicalChannel`s that are referencing a `CommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CommunicationConnector` shall only be referenced by at most one `PhysicalChannel`.⌋

[constr_3378] Maximal one AliasNameAssignment allowed per FlatInstanceDescriptor

Imposition time: CP: IT_EcuExt

⌈In a given instance of AliasNameSet in the bound system there shall be at most one aliasName per FlatInstanceDescriptor.⌋

[constr_3379] Multiple SocketAddress entries with the same IP Address, Protocol and Port in the context of a given EcuInstance

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3383] Standardized values for the attribute category of meta-class GeneralPurposeConnection

Imposition time: CP: IT_SysDesc

⌈The following values of the attribute category of meta-class GeneralPurposeConnection are reserved by the AUTOSAR standard:

- XcpChannel

⌋

[constr_3384] PduTriggerings referenced by GeneralPurposeConnection shall be defined on the same PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈The PduTriggerings that are referenced by the GeneralPurposeConnection in the role pduTriggering shall be defined on the same PhysicalChannel.⌋

[constr_3385] XcpChannel is allowed to reference exactly two PduTriggerings

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3399] Existence of securedAreaOffset and securedAreaLength

Imposition time: CP: IT_SysDesc

⌈If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.⌋

[constr_3402] Mandatory offset if noHeader is used

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = noHeader the ContainedIPduProps.offset shall be defined.⌋

[constr_3403] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used

Imposition time: CP: IT_SysDesc

⌈If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.⌋

[constr_3404] Usage of ContainedIPduProps.updateIndicationBitPosition

Imposition time: CP: IT_SysDesc

⌈ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering is set to noHeader.⌋

[constr_3405] Dynamic Length IPdu inside of a static configured ContainerIPdu

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3406] All signals before authDataFreshnessStartPosition shall have a static length

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3407] Freshness Value in Authentic IPdu is not allowed to be used in case of ContainerIPdu with a dynamic layout

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3436] Value range of minimumTxContainerQueueSize and minimumRxContainerQueueSize

Imposition time: CP: IT_SysDesc

⌈If defined, the value of minimumTxContainerQueueSize and minimumRxContainerQueueSize shall be in the range of 0..255.⌋

[constr_3437] invalidValue defined in the context of ISignal

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3448] Restriction for usage of Pdu.hasDynamicLength

Imposition time: CP: IT_SysDesc

⌈The Pdu.hasDynamicLength attribute is only relevant for UserDefinedPdus, UserDefinedIPdus, J1939DcmIPdus.⌋

[constr_3454] Unique headerIdLongHeader for acceptConfigured

Imposition time: CP: IT_SysDesc

⌈For a ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured and ContainerIPdu.headerType = longHeader the following shall apply: All referenced IPdus (via ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering) shall have a unique ContainedIPduProps.headerIdLongHeader within the scope of this ContainerIPdu.⌋

[constr_3455] Unique headerIdShortHeader for acceptConfigured

Imposition time: CP: IT_SysDesc

⌈For a ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured and ContainerIPdu.headerType = shortHeader the following shall apply: All referenced IPdus (via ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering) shall have a unique ContainedIPduProps.headerIdShortHeader within the scope of this ContainerIPdu.⌋

[constr_3456] Existence of ProvidedServiceInstance.loadBalancingPriority and ProvidedServiceInstance.loadBalancingWeight

Imposition time: CP: IT_SysDesc

⌈The attributes ProvidedServiceInstance.loadBalancingPriority and ProvidedServiceInstance.loadBalancingWeight shall either not exist or be defined both.⌋

[constr_3457] Uniqueness of ConsumedEventGroup.eventGroupIdentifier in the scope of a ConsumedServiceInstance

Imposition time: CP: IT_SysDesc

⌈Each ConsumedEventGroup that is aggregated by a ConsumedServiceInstance shall have a unique eventGroupIdentifier value in the scope of the aggregating ConsumedServiceInstance.⌋

[constr_3458] FlatInstanceDescriptor.rtePluginProps shall only reference a EcucContainerValue representing a RteRipsPlugin

Imposition time: CP: IT_EcuExt

⌈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⌋

[constr_3460] Full definition of `transferProperty` for group signal

Imposition time: CP: `IT_SysDesc`

⌈If at least one of the `ISignal`s belonging to an `ISignalGroup` has a `transferProperty` defined (via their respective `ISignalToIPduMapping`) then
all other `ISignal`s belonging to the same `ISignalGroup` shall have a `transferProperty` defined as well.⌋

[constr_3461] TransferProperty for group signals if `ISignalGroup` has `transferProperty=pending`

Imposition time: CP: `IT_SysDesc`

⌈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.

⌋

[constr_3464] Allowed Pdu type on `BusMirrorChannelMapping.targetChannel`

Imposition time: CP: `IT_SysDesc`

⌈Each `PduTriggering` that is referenced by `BusMirrorChannelMapping` in the role `targetPduTriggering` is only allowed to reference a `GeneralPurposeIPdu` of category BUS_MIRRORING.⌋

[constr_3465] Identical `BusMirrorChannel.busMirrorNetworkId` for `BusMirrorChannel`s referencing the same `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `BusMirrorChannel.busMirrorNetworkId` shall be identical in all `BusMirrorChannel`s that are referencing the same `PhysicalChannel` in the scope of the `System`.⌋

[constr_3466] Unique `BusMirrorChannel.busMirrorNetworkId`s for each specialization of `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `BusMirrorChannel.busMirrorNetworkId` associated with `PhysicalChannel`s that have the same specialization (e.g. all `CanPhysicalChannel`s) shall have unique `BusMirrorChannel.busMirrorNetworkId`s within the scope of the `System`).⌋

[constr_3467] `CanPhysicalChannel` as destination channel of `BusMirrorChannelMappingCan`

Imposition time: CP: `IT_SysDesc`

⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingCan` shall only reference a `CanPhysicalChannel` in the role `targetChannel`.⌋

[constr_3468] `BusMirrorChannelMappingCan.targetPduTriggering` restriction

Imposition time: CP: `IT_SysDesc`

⌈`BusMirrorChannelMappingCan` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.⌋

[constr_3469] `CanFrameTriggering.txMask` setting for the destination frame

Imposition time: CP: `IT_SysDesc`

⌈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_3470] PaddingValue used to transmit the Pdu on a Can-Fd destination bus

Imposition time: CP: `IT_SysDesc`

⌈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.

⌋

[constr_3471] `FlexrayPhysicalChannel` as destination channel of `BusMirrorChannelMappingFlexray`

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3474] EthernetPhysicalChannel as destination channel of BusMirrorChannelMappingIp

Imposition time: CP: IT_SysDesc

⌈The BusMirrorChannel that is aggregated by BusMirrorChannelMappingIp shall only reference an EthernetPhysicalChannel in the role targetChannel.⌋

[constr_3475] BusMirrorChannelMappingIp.targetPduTriggering restriction

Imposition time: CP: IT_SysDesc

⌈BusMirrorChannelMappingIp is allowed to reference only one single PduTriggering in the role targetPduTriggering.⌋

[constr_3476] UserDefinedPhysicalChannel as destination channel of BusMirrorChannelMappingUserDefined

Imposition time: CP: IT_SysDesc

⌈The BusMirrorChannel that is aggregated by BusMirrorChannelMappingUserDefined shall only reference a UserDefinedPhysicalChannel in the role targetChannel.⌋

[constr_3477] BusMirrorChannelMappingUserDefined.targetPduTriggering restriction

Imposition time: CP: IT_SysDesc

⌈BusMirrorChannelMappingUserDefined is allowed to reference only one single PduTriggering in the role targetPduTriggering.⌋

[constr_3479] `PhysicalChannel` is not allowed to be a `managedPhysicalChannel` and a managing `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈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_3484] `PncMapping` that refers a `managedPhysicalChannel` shall also refer the managing `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈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`).⌋

[constr_3488] Value range of `ContainedIPduProps.priority`

Imposition time: CP: `IT_SysDesc`

⌈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

Imposition time: CP: IT_SysDesc

⌈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`

Imposition time: CP: IT_SysDesc

⌈ContainedIPduProps.priority is only applicable if ContainedIPduProps.collectionSemantics is set to lastIsBest.⌋

[constr_3501] Role of `SystemSignal` in 1:n communication

Imposition time: CP: IT_EcuExt

⌈In case of 1:n communication the VariableDataPrototype in the PPortPrototype of the SwComponentPrototype shall be mapped to only one SystemSignal.⌋

[constr_3508] Value of `nmReadySleepTime`

Imposition time: CP: IT_SysDesc

⌈The nmReadySleepTime value shall be a multiple of cycle * nmRepetitionCycle.⌋

[constr_3514] No two `ISignalToIPduMapping`s shall reference the identical `ISignal`

Imposition time: CP: IT_SysDesc

⌈No two ISignalToIPduMappings shall reference the identical ISignal in the role iSignal in the scope of one System.⌋

[constr_3515] Fully filled `EthernetPriorityRegeneration` table

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈In case the CouplingPortDetails.ethernetPriorityRegeneration is defined it shall contain exactly 8 elements of EthernetPriorityRegeneration, one for each value of ingressPriority (0-7).⌋

[constr_3516] limitation of `Frame.frameLength` for CAN L-PDUs

Imposition time: CP: IT_SysDesc

⌈The Frame.frameLength of CAN PDUs shall be restricted to

- 0..8 for classic CAN L-PDUs;
- 0..8, 12, 16, 20, 24, 32, 48, 64 for CAN FD L-PDUs and
- 1..2048 for CAN XL L-PDUs.

⌋

[constr_3517] Consistent setting of ContainedIPduProps.collectionSemantics in the context of one ContainerIPdu

Imposition time: CP: IT_SysDesc

⌈The value of the attribute ContainedIPduProps.collectionSemantics shall be identical for all contained IPdus within the context of a given ContainerIPdu.⌋

[constr_3518] Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Imposition time: CP: IT_SysDesc

⌈The value given for CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue shall be in the range from 0 to 255.⌋

[constr_3519] Value of category of GlobalTimeDomain

Imposition time: CP: IT_SysDesc

⌈The attribute category of GlobalTimeDomain can have the following value:

- SYNCHRONIZED: this time base does not depend on the existence of another time base

⌋

[constr_3521] defaultVlan and vlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈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_3523] CouplingPort and PncMapping in the scope of an EthernetPhysicalChannel

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3524] Definition of couplingPortRole on CouplingPort for managed CouplingElement

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈A CouplingPort with couplingPortRole set to upLinkPort shall be connected to exactly one other CouplingPort with couplingPortRole set to upLinkPort.⌋

[constr_3533] EndToEndTransformationISignalProps.dataLength shall be a multiple of 8

Imposition time: CP: IT_SysDesc

⌈The value of EndToEndTransformationISignalProps.dataLength, EndToEndTransformationISignalProps.maxDataLength, and EndToEndTransformationISignalProps.minDataLength shall be a multiple of 8.⌋

[constr_3534] EthernetPhysicalChannel shall only be referenced by one VlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈An EthernetPhysicalChannel shall only be referenced by one VlanMembership in the role VlanMembership.vlan in the scope of one CouplingPort.⌋

[constr_3535] EthernetCommunicationController shall aggregate at most one CouplingPort

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈An EthernetCommunicationController is allowed to aggregate at most one CouplingPort.⌋

[constr_3545] Mandatory reference to a *Pnc* in case of anyPartialNetworkActive or allPartialNetworksActive

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationProps.serviceControl equals anyPartialNetworkActive or allPartialNetworksActive, then the reference SignalServiceTranslationProps.controlPnc shall point to at least one PncMappingIdent.⌋

[constr_3546] Mandatory reference to a ConsumedEventGroup in case of serviceControl

Imposition time: CP: IT_SysDesc

⌈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_3548] EndToEnd profile for both ends of safeTranslation

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3549] Secure payload for both ends in case of secureTranslation

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_3559] ConsumedServiceInstance.blocklistedVersion is restricted to the usage of minorVersion

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The majorVersion attribute shall not be used in the SomeipServiceVersion that is aggregated by the ConsumedServiceInstance in the role blocklistedVersion.⌋

[constr_3560] minimumMinorVersion and ConsumedServiceInstance.minorVersion value

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.minorVersion shall not have the value *ANY* if versionDrivenFindBehavior = minimumMinorVersion.⌋

[constr_3600] Setting of EthernetCommunicationController.slaveActAsPassiveCommunicationSlave

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute EthernetCommunicationController.slaveActAsPassiveCommunicationSlave may only be set to TRUE, if the following conditions apply:

- the EthernetCommunicationController is not referenced by any NmNode in the role controller

- the EthernetCommunicationController aggregates at least one CouplingPort
- the couplingPortRole of that CouplingPort is set to standardPort
- the physicalLayerType of that CouplingPort is set to either _100BASE_T1, _1000BASE_T1 or _10BASE_T1S

In all other cases the attribute slaveActAsPassiveCommunicationSlave shall be set to FALSE or shall not be defined.⌋

[constr_3601] Mandatory attributes of EthernetWakeupSleepOnDataline-Config

Imposition time: CP: IT_EcuExt, AP: IT_SysDes

⌈The following attributes of EthernetWakeupSleepOnDatalineConfig shall be defined:

- wakeupLocalEnabled
- wakeupRemoteEnabled

⌋

[constr_3602] Existence of wakeupForwardLocalEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupForwardLocalEnabled shall be defined if wakeupRemote-Enabled is set to TRUE.⌋

[constr_3603] Existence of wakeupLocalDurationTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupLocalDurationTime shall be defined if wakeupForward-LocalEnabled is set to TRUE.⌋

[constr_3604] Existence of wakeupForwardRemoteEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupForwardRemoteEnabled shall be defined if wakeupLocal-Enabled is set to TRUE.⌋

[constr_3605] Existence of wakeupLocalDetectionTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupLocalDetectionTime shall be defined if wakeupForward-RemoteEnabled is set to TRUE.⌋

[constr_3606] Values of wakeupLocalDurationTime and wakeupLocalDetectionTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, then the value of wakeupLocalDurationTime shall be greater than the value of wakeupLocalDetectionTime.⌋

[constr_3607] Existence of sleepRepetitionDelayOfSleepRequest

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute sleepRepetitionDelayOfSleepRequest shall be defined if sleepRepetitionsOfSleepRequest is defined and has a value greater than 0.⌋

[constr_3608] Existence of wakeupRepetitionDelayOfWakeupRequest

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupRepetitionDelayOfWakeupRequest shall only be defined if wakeupRepetitionsOfWakeupRequest is defined and has a value greater than 0.⌋

[constr_3609] Values of wakeupLocalDurationTime in the context of a CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈All CouplingPorts which have the reference wakeupSleepOnDatalineConfig defined and

- where the CouplingPorts are aggregated by the same CouplingElement and
- where the referenced EthernetWakeupSleepOnDatalineConfig has the attribute wakeupLocalDurationTime defined

shall refer to EthernetWakeupSleepOnDatalineConfigs where the value of wakeupLocalDurationTime is identical for all referencing CouplingPorts.⌋

[constr_3610] Values of wakeupLocalDetectionTime in the context of a CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈All CouplingPorts which have the reference wakeupSleepOnDatalineConfig defined and

- where the CouplingPorts are aggregated by the same CouplingElement and
- where the referenced EthernetWakeupSleepOnDatalineConfig has the attribute wakeupLocalDetectionTime defined

shall refer to EthernetWakeupSleepOnDatalineConfigs where the value of wakeupLocalDetectionTime is identical for all referencing CouplingPorts.⌋

[constr_3611] Existence of EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute slaveQualifiedUnexpectedLinkDownTime shall be defined if slaveActAsPassiveCommunicationSlave is set to TRUE.⌋

[constr_3615] Existence of EthernetCluster.couplingPortSwitchoffDelay

Imposition time: CP: IT_SysDesc

⌈The attribute EthernetCluster.couplingPortSwitchoffDelay shall be defined if at least one EcuInstance connected to that EthernetCluster has the attribute ethSwitchPortGroupDerivation set to TRUE.⌋

[constr_3616] Value of EthernetCluster.couplingPortSwitchoffDelay

Imposition time: CP: IT_SysDesc

⌈If defined, the value of EthernetCluster.couplingPortSwitchoffDelay shall be greater than UdpNmCluster.nmNetworkTimeout + UdpNmCluster.nmWaitBusSleepTime of the respective EthernetCluster.⌋

[constr_3617] Existence of EthernetCluster.couplingPortStartupActiveTime

Imposition time: CP: IT_SysDesc

⌈The attribute EthernetCluster.couplingPortStartupActiveTime shall be defined if at least one EcuInstance connected to that EthernetCluster has the attribute ethSwitchPortGroupDerivation set to TRUE.⌋

[constr_3618] Value of EthernetCluster.couplingPortStartupActiveTime

Imposition time: CP: IT_SysDesc

⌈If defined, the value of EthernetCluster.couplingPortStartupActiveTime shall be greater than UdpNmCluster.nmNetworkTimeout + UdpNmCluster.nmWaitBusSleepTime of the respective EthernetCluster.⌋

[constr_3620] GlobalTimeDomain.networkSegmentId only applicable to Global Time sub domains

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The aggregation GlobalTimeDomain.networkSegmentId shall only be defined if the GlobalTimeDomain is itself referenced in the role GlobalTimeDomain.globalTimeSubDomain.⌋

[constr_3621] Value range of GlobalTimeDomain.networkSegmentId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of GlobalTimeDomain.networkSegmentId shall be in the range 0..255.⌋

[constr_3651] No `element` in case `translationTarget` is primitive

Imposition time: CP: IT_SysDesc

⌈If SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a primitive AutosarDataType then the reference SignalServiceTranslationElementProps.element shall not be defined.⌋

[constr_3652] Allowed sub-classes of `DataPrototypeReference` in the context of `signal/service translation`

Imposition time: CP: IT_SysDesc

⌈If a DataPrototypeReference in the role SignalServiceTranslationElementProps.element is used then following sub-classes are supported:

- if the reference target is typed by an ApplicationDataType then the DataPrototypeInSenderReceiverInterfaceInstanceRef shall be used and shall target an ApplicationCompositeElementDataPrototype.
- if the reference target is typed by an ImplementationDataType then the ImplementationDataTypeElementInPortInterfaceRef shall be used.

⌋

[constr_3653] Consistent `translationTarget` and `element` in case `ApplicationDataType` is used

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by an ApplicationDataType (targetDataPrototype of the VariableDataPrototypeInSystemInstanceRef) then every SignalServiceTranslationElementProps.element reference that is defined in the context of the SignalServiceTranslationEventProps shall have that VariableDataPrototype as the rootDataPrototypeInSr of the DataPrototypeInSenderReceiverInterfaceInstanceRef.⌋

[constr_3654] Consistent `translationTarget` and `element` in case `ImplementationDataType` is used

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by an ImplementationDataType (targetDataPrototype of the VariableDataPrototypeInSystemInstanceRef) then every SignalServiceTranslationElementProps.element reference that is defined in the context of the SignalServiceTranslationEventProps shall have that VariableDataPrototype as the rootDataPrototype of the ImplementationDataTypeElementInPortInterfaceRef.⌋

[constr_3655] Supported filter types for primitive SignalServiceTranslationElementProps

Imposition time: CP: IT_SysDesc

⌈If the target for SignalServiceTranslationElementProps is defined as primitive according to [TPS_SYST_03062] then the following values for dataFilterType are supported:

- always
- maskedNewDiffersMaskedOld
- maskedNewDiffersX
- maskedNewEqualsX
- never
- newIsOutside
- newIsWithin
- oneEveryN.

⌋

[constr_3656] Supported filter types for composite SignalServiceTranslationElementProps

Imposition time: CP: IT_SysDesc

⌈If the target for SignalServiceTranslationElementProps is defined as composite according to [TPS_SYST_03062] then the following values for dataFilterType are supported:

- always
- never
- oneEveryN.

⌋

[constr_3668] Existence of TlsCryptoCipherSuite.cipherSuiteShortLabel

Imposition time: CP: IT_SysDesc

⌈If a TlsCryptoCipherSuite.cipherSuiteShortLabel is defined then:

- the attribute TlsCryptoCipherSuite.cipherSuiteId shall be defined as well
- the value of TlsCryptoCipherSuite.cipherSuiteShortLabel shall match the *Description* value corresponding to the *Value* field defined in TlsCryptoCipherSuite.cipherSuiteId according to TlsCryptoCipherSuite Parameter set defined in [10].

⌋

[constr_3669] `eventMulticastSubscriptionAddress` shall refer to a multicast address

Imposition time: CP: `IT_SysDesc`

⌈The reference `ConsumedServiceInstance.eventMulticastSubscriptionAddress` shall refer to an `ApplicationEndpoint` which in turn refers to a `NetworkEndpoint` that represents a multicast address.⌋

[constr_3670] No support for parallel `localUnicastAddress` and `eventMulticastSubscriptionAddress`

Imposition time: CP: `IT_SysDesc`

⌈If a `eventMulticastSubscriptionAddress` is defined for a `ConsumedServiceInstance` then there shall not be a `localUnicastAddress` defined at the same `ConsumedServiceInstance`.⌋

[constr_3671] `remoteMulticastSubscriptionAddress` shall refer to a multicast address

Imposition time: CP: `IT_SysDesc`

⌈The reference `ProvidedServiceInstance.remoteMulticastSubscriptionAddress` shall refer to an `ApplicationEndpoint` which in turn refers to a `NetworkEndpoint` that represents a multicast address.⌋

[constr_3672] No support for methods in `multicast subscription` at the client

Imposition time: CP: `IT_SysDesc`

⌈If a `ConsumedServiceInstance` aggregates a `PduActivationRoutingGroup` in the role `methodActivationRoutingGroup`, then the `ConsumedServiceInstance` shall not define a `eventMulticastSubscriptionAddress`.⌋

[constr_3673] No support for methods in `multicast subscription` at the server static configuration

Imposition time: CP: `IT_SysDesc`

⌈If a `ProvidedServiceInstance` aggregates a `PduActivationRoutingGroup` in the role `methodActivationRoutingGroup`, then the `ProvidedServiceInstance.remoteMulticastSubscriptionAddress` shall not be defined.⌋

[constr_3685] Allowed values for each element of `pncFilterArrayMask`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The value for each element of `CommunicationConnector.pncFilterArrayMask` shall be in the range between 0 and 255.⌋

[constr_3686] Allowed number of entries for pncFilterArrayMask

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The number of CommunicationConnector.pncFilterArrayMask elements shall be:

- NmCluster.pncClusterVectorLength, if defined
- System.pncVectorLength, otherwise.

⌋

[constr_3687] Limited value range for NmCluster.pncClusterVectorLength

Imposition time: CP: IT_SysDesc

⌈The value of NmCluster.pncClusterVectorLength shall be equal or smaller than System.pncVectorLength.⌋

[constr_3695] canControllerXlAttributes and canControllerXlRequirements are mutually exclusive

Imposition time: CP: IT_SysDesc

⌈The existence of canControllerXlAttributes and canControllerXlRequirements is mutually exclusive.⌋

[constr_3696] Mandatory attributes of CanControllerXlConfiguration

Imposition time: CP: IT_SysDesc

⌈A CanControllerConfiguration configuring a CAN XL controller shall aggregate CanControllerXlConfiguration with the following attributes defined:

- errorSignalingEnabled
- propSeg
- syncJumpWidth
- timeSeg1
- timeSeg2
- trcvPwmModeEnabled

⌋

[constr_3697] Latest existence time of CanControllerXlConfiguration and CanControllerXlConfigurationRequirements

Imposition time: CP: IT_SysDesc

⌈In case that a CAN XL controller is configured, then either CanControllerXlConfiguration or CanControllerXlConfigurationRequirements shall exist within their aggregating class CanControllerConfiguration or CanControllerConfigurationRequirements.⌋

[constr_3698] Value of errorSignalingEnabled

Imposition time: CP: IT_SysDesc

⌈The attribute errorSignalingEnabled shall be set to FALSE if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3699] Existence of pwmL

Imposition time: CP: IT_SysDesc

⌈The attribute pwmL shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3700] Existence of pwmO

Imposition time: CP: IT_SysDesc

⌈The attribute pwmO shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3701] Existence of pwmS

Imposition time: CP: IT_SysDesc

⌈The attribute pwmS shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3702] Relevant attributes of EthernetCommunicationController for CAN XL

Imposition time: CP: IT_SysDesc

⌈If the category of EthernetCommunicationController is equal to CAN_XL, then only the following attributes of this meta-class are relevant:

- macLayerType
- macUnicastAddress

⌋

[constr_3703] Reference to CanControllerXlConfiguration in case of category CAN_XL

Imposition time: CP: IT_SysDesc

⌈If the category of EthernetCommunicationController is equal to CAN_XL, then the reference canXlConfig of EthernetCommunicationController shall refer to the CanCommunicationController aggregating the CanControllerConfiguration which in turn aggregates the CanControllerXlConfiguration that is used for tunneling of the Ethernet frames associated with the aforementioned EthernetCommunicationController.⌋

[constr_3704] Existence of CanXlFrameTriggeringProps

Imposition time: CP: IT_SysDesc

⌈If the class CanXlFrameTriggeringProps is aggregated by a CanFrameTriggering, then the CanCommunicationController – which is referenced through commController by a CanCommunicationConnector which in turn is ref-

erenced through commConnector by a CanPhysicalChannel that aggregates the aforementioned CanFrameTriggering – shall aggregate at least one of

- CanControllerConfiguration with CanControllerXlConfiguration aggregated or
- CanControllerConfigurationRequirements with CanControllerXl-ConfigurationRequirements aggregated.

⌋

[constr_3705] Allowed values for priorityId

Imposition time: CP: IT_SysDesc

⌈The value for priorityId shall be in the range between 0 and 2047.⌋

[constr_3706] Allowed values for sduType

Imposition time: CP: IT_SysDesc

⌈The value for sduType shall be in the range between 0 and 255.⌋

[constr_3707] Allowed values for vcid

Imposition time: CP: IT_SysDesc

⌈The value for vcid shall be in the range between 0 and 255.⌋

[constr_3708] No UDP network management in case of Ethernet tunneling through CAN XL

Imposition time: CP: IT_SysDesc

⌈For an EthernetPhysicalChannel that is connected to an Ethernet-CommunicationController of category CAN_XL (i.e. an EthernetPhysical-Channel tunneled through CAN XL), no UDP network management shall be configured.⌋

[constr_3713] Allowed values for acceptanceField

Imposition time: CP: IT_SysDesc

⌈The value for acceptanceField shall be in the range between 0 and 4294967295.⌋

[constr_3714] Multiple top level PNC-coordinators shall be allowed

Imposition time: CP: IT_SysDesc

⌈Multiple top level PNC-coordinators shall only be allowed if no network path across all networks exist that connects a CommunicationConnector with pncGateway-Type PncGatewayTypeEnum.active to another CommunicationConnector with pncGatewayType PncGatewayTypeEnum.active where both Communication-Connectors belong to different top level PNC-coordinators.⌋

[constr_3716] `SecuredIPdu.dynamicRuntimeLengthHandling` for dynamic length `Pdu`s

Imposition time: CP: `IT_SysDesc`

⌈If a `PduTriggering` is referenced from a `SecuredIPdu` in the role `payload` and the `Pdu` referenced by the `PduTriggering` in the role `iPdu` qualifies according to [TPS_SYST_03085] to be of dynamic length, then the `SecuredIPdu` shall have the attribute `SecuredIPdu.dynamicRuntimeLengthHandling` set to true.⌋

[constr_3717] `SecuredIPdu.dynamicRuntimeLengthHandling` for gateway operation with `IPduMapping.pduMaxLength` defined

Imposition time: CP: `IT_SysDesc`

⌈If a `PduTriggering` refers to a `SecuredIPdu` in the role `iPdu` and that `PduTriggering` is used in an `IPduMapping` where a `pduMaxLength` value is defined (either in the role `IPduMapping.sourceIPdu` or `TargetIPduRef.targetIPdu`), then the `SecuredIPdu` shall have the attribute `SecuredIPdu.dynamicRuntimeLengthHandling` set to true.⌋

[constr_3718] Minimum length of `SecuredIPdu`s

Imposition time: CP: `IT_SysDesc`

⌈If a `SecuredIPdu` has the attribute `useAsCryptographicIPdu` set to false, then the `length` attribute of that `SecuredIPdu` shall be at least the sum of the `payload Pdu.length` and `SecuredIPdu.authenticationProps.authInfoTxLength`.⌋

[constr_3726] Upper multiplicity of aggregation in the role `CouplingPort.macSecProps`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈In the context of `CouplingPort`, the aggregation in the role `macSecProps` shall exist at most once.⌋

[constr_3735] Existence of `DdsCpServiceInstance.ddsServiceQosProfile`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the reference in the role `ddsServiceQosProfile` shall exist.

Note: This profile applies to the all 4 topics used for methods and fields defined into the `DdsCpServiceInstance` (e.g `ddsMethodRequestTopic`, `ddsMethodReplyTopic`), `ddsFieldRequestTopic`, `ddsFieldReplyTopic`).⌋

[constr_3736] `ISignal` that has `dataTypePolicy` set to `ddsSignal` shall be referenced by a `DdsCpISignalToDdsTopicMapping`

Imposition time: CP: `IT_SysDesc`

⌈Every `ISignal` that has `dataTypePolicy` set to `ddsSignal` shall be referenced by a `DdsCpISignalToDdsTopicMapping`.⌋

[constr_3737] `ISignal` referenced from `DdsCpISignalToDdsTopicMapping`

Imposition time: CP: `IT_SysDesc`

⌈Every `ISignal` that has `dataTypePolicy` set to any value different to `ddsSignal` shall NOT be referenced by a `DdsCpISignalToDdsTopicMapping`.⌋

[constr_3739] Value of `ISignal.dataTypePolicy` for all `ISignal`s associated with a `DdsCpServiceInstance`

Imposition time: CP: `IT_EcuExt`

⌈For all `ISignal`s that are referenced by an `ISignalToIPduMapping` where the enclosing `ISignalIPdu` is only referenced by `PduTriggering`s that are in turn referenced in one of the roles:

- `DdsServiceInstanceEventCp.eventTriggering`
- `DdsServiceInstanceOperationCp.operationRequestTriggering`
- `DdsServiceInstanceOperationCp.operationResponseTriggering`

The value of attribute `ISignal.dataTypePolicy` shall be set to `ddsService`.⌋

[constr_3740] Existence of `DdsServiceInstanceEventCp.ddsEventTopic`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsServiceInstanceEventCp`, the reference in the role `ddsEventTopic` shall exist.⌋

[constr_3741] Exclusive setting of `channelSynchronousWakeup` or `pncSynchronousWakeup`

Imposition time: CP: `IT_SysDesc`

⌈At most one of `EcuInstance.channelSynchronousWakeup` or `EcuInstance.pncSynchronousWakeup` shall be set to TRUE.⌋

[constr_3742] Value for `createEcuWakeupSource` in the conntext of a `CommunicationCluster`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `CommunicationConnector.createEcuWakeupSource` shall be set to the same value for all `CommunicationConnector`s in the scope of one `EcuInstance` which are referenced by `PhysicalChannel`s that belong to the same `CommunicationCluster`.⌋

[constr_3743] Allowed values for `IEEE1722TpConnection.uniqueStreamId`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The value for `IEEE1722TpConnection.uniqueStreamId` shall be in the range between 0 and 65535.⌋

[constr_3744] Allowed values for `IEEE1722TpConnection.version`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The value for `IEEE1722TpConnection.version` shall be in the range between 0 and 7.⌋

[constr_3745] `category` of `GeneralPurposePdu` referenced in the role `IEEE1722TpConnection.pdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈The `GeneralPurposePdu` referenced by the `PduTriggering` which in turn is referenced in the role `pdu` by the `IEEE1722TpConnection` shall have the `category` set to `IEEE1722TP`.⌋

[constr_3746] `category` of `GeneralPurposePdu` referenced in the role `IEEE1722TpAvConnection.sdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈The `GeneralPurposePdu` referenced by the `PduTriggering` which in turn is referenced in the role `IEEE1722TpAvConnection.sdu` by the `IEEE1722TpAvConnection` shall have the `category` set to `IEEE1722TP_STREAM`.⌋

[constr_3747] Existence of attribute `IEEE1722TpConnection.uniqueStreamId`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpConnection`, the attribute `uniqueStreamId` shall exist.⌋

[constr_3748] Existence of attribute `IEEE1722TpConnection.macAddressStreamId`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpConnection`, the attribute `macAddressStreamId` shall exist.⌋

[constr_3749] Existence of attribute `IEEE1722TpConnection.version`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpConnection`, the attribute `version` shall exist.⌋

[constr_3750] Existence of attribute `IEEE1722TpConnection.pdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈For each `IEEE1722TpConnection`, the reference `pdu` shall exist.⌋

[constr_3751] Allowed values for `IEEE1722TpAcfBus.busId`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The value for `IEEE1722TpAcfBus.busId` shall be in the range between 0 and 31.⌋

[constr_3752] Existence of attribute `IEEE1722TpAcfBus.busId`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpAcfBus`, the attribute `busId` shall exist.⌋

[constr_3753] Existence of attribute `IEEE1722TpAcfLinPart.linIdentifier`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpAcfLinPart`, the attribute `linIdentifier` shall exist.⌋

[constr_3754] Existence of attribute `IEEE1722TpAcfCan.messageType`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈For each `IEEE1722TpAcfCan`, the attribute `messageType` shall exist.⌋

[constr_3755] Consistend aggregation of `IEEE1722TpAcfCanPart`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈An `IEEE1722TpAcfCan` shall only aggregate `IEEE1722TpAcfCanPart`s in the role `IEEE1722TpAcfCan.acfPart`.⌋

[constr_3756] Consistend aggregation of `IEEE1722TpAcfLinPart`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈An `IEEE1722TpAcfLin` shall only aggregate `IEEE1722TpAcfLinPart`s in the role `IEEE1722TpAcfLin.acfPart`.⌋

[constr_3757] Allowed values for `IEEE1722TpAcfLinPart.linIdentifier`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The value for `IEEE1722TpAcfLinPart.linIdentifier` shall be in the range between 0 and 63.⌋

[constr_3758] Allowed values for `IEEE1722TpAcfCanPart.canIdentifier`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The value for `IEEE1722TpAcfCanPart.canIdentifier` shall be in the range between 0 and 536870911.⌋

[constr_3759] Existence of attribute `IEEE1722TpAcfCanPart.canIdentifier` for `IEEE1722Tp` ACF stream transmission

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈If an `IEEE1722TpAcfCanPart` is part of an `IEEE1722TpAcfConnection` which, according to [TPS_SYST_03109], is transmitted in that `IEEE1722TpAcfConnection` and

the `IEEE1722TpAcfCanPart` refers to a `PduTriggering` in the role `sdu` which in turn refers to a `Pdu` that is NOT a `GeneralPurposePdu` of `category` `IEEE1722TP_ID_RANGE`,

then the attribute `IEEE1722TpAcfCanPart.canIdentifier` shall exist.⌋

[constr_3760] Existence of attribute `IEEE1722TpAcfCanPart.canIdentifierRange` or `canIdentifierMask` for `IEEE1722Tp` ACF stream reception

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈If an `IEEE1722TpAcfCanPart` is part of an `IEEE1722TpAcfConnection` which, according to [TPS_SYST_03110], is received in that `IEEE1722TpAcfConnection` and

the `IEEE1722TpAcfCanPart` refers to a `PduTriggering` in the role `sdu` which in turn refers to a `Pdu` that is NOT a `GeneralPurposePdu` of `category` `IEEE1722TP_ID_RANGE`,

then the attribute `IEEE1722TpAcfCanPart.canIdentifierRange` or `canIdentifierMask` shall exist.⌋

[constr_3761] Identical `EthernetPhysicalChannel` owning `PduTriggering`s referenced by `IEEE1722TpConnection.pdu` and `IEEE1722TpAvConnection.sdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈The `PduTriggering`s referenced in the roles `IEEE1722TpConnection.pdu` and `IEEE1722TpAvConnection.sdu` shall be owned by the same `EthernetPhysicalChannel`.⌋

[constr_3762] Usage of `CouplingElementSwitchDetails` only on an Ethernet switch

Status: DRAFT
Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingElement` aggregates a `CouplingElementSwitchDetails` in the role `CouplingElement.couplingElementDetails`, then that `CouplingElement` shall have the attribute `couplingType` set to the value `CouplingElementEnum.switch`.⌋

[constr_3763] Allowed value for `maxDeltaCounter` in the context of a `profileName`

Imposition time: CP: `IT_SysDesc`

⌈An `EndToEndTransformationComSpecProps` that is associated with an `EndToEndTransformationDescription` as described in [TPS_SYST_02275] shall not contain a `maxDeltaCounter` value that is outside the value range imposed by the profile defined in `EndToEndTransformationDescription.profileName`.

The profile specific value ranges are listed in [constr_3158], [constr_3195], [constr_3159], [constr_3196], [constr_3197], and [constr_3316].⌋

[constr_3764] Applicability of `CouplingPort.macAddressVlanAssignment`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The aggregation `CouplingPort.macAddressVlanAssignment` shall only exist if the `CouplingPort` is aggregated by a `CouplingElement` with `couplingType` = `CouplingElementEnum.switch`.⌋

[constr_3765] Applicability of `MacAddressVlanMembership.vlan`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The reference `MacAddressVlanMembership.vlan` shall only exist if the `CouplingPort` aggregating this `MacAddressVlanMembership` is aggregated by a `CouplingElement` with `switchMacAddressLearningMode` = `SwitchMacAddressLearningEnum.independentVlanLearning`.⌋

[constr_3766] Valid `MacAddressVlanMembership.vlan` target `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If an `EthernetPhysicalChannel` is referenced by a `CouplingPort.macAddressVlanAssignment.vlan`, then that `EthernetPhysicalChannel` shall also be referenced by the same `CouplingPort` via `CouplingPort.vlanMembership.vlan`⌋

[constr_3767] `CommunicationConnector.nmVariant` setting to `slavePassive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.slavePassive` if an `EthernetCommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `slaveActAsPassiveCommunicationSlave` in the referenced `EthernetCommunicationController` is set to true.⌋

[constr_3768] `CommunicationConnector.nmVariant` setting to `slaveActive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.slaveActive` if a `LinSlave` is referenced in the role `EcuInstance.commController`.⌋

[constr_3769] `CommunicationConnector.nmVariant` setting to `full`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.full` if a `CommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `nmPassiveModeEnabled` in the `NmEcu` that is referencing this `CommunicationController` is not present or is set to false.⌋

[constr_3770] `CommunicationConnector.nmVariant` setting to `passive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.passive` if a `CommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `nmPassiveModeEnabled` in the `NmEcu` that is referencing this `CommunicationController` is set to true.⌋

[constr_3771] Range of `NmCluster.nmLightTimeout`

Imposition time: CP: `IT_SysDesc`

⌈The value given for `NmCluster.nmLightTimeout` shall be in the range from 0 to 255.⌋

[constr_3781] Each PNC assigned to multiple `PhysicalChannel`s shall have a top level PNC-Coordinator

Imposition time: CP: `IT_SysDesc`

⌈In a `System`, if a PNC is assigned to multiple `PhysicalChannel`s according to [TPS_SYST_03080], then this PNC shall have at least one top level PNC-Coordinator according to [TPS_SYST_03082].⌋

[constr_3782] Consistent `framePreemptionSupport` setting in the scope of one `CouplingPortConnection`

Imposition time: CP: `IT_SysDesc`

⌈For each `CouplingPortConnection` the value of `CouplingPort.couplingPortDetails.framePreemptionSupport` shall be identical for both, `CouplingPortConnection.firstPort` and `CouplingPortConnection.secondPort`.⌋

[constr_3783] Definition of `CouplingPortFifo.trafficClassPreemptionSupport` only in context of an Ethernet switch

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingPort` is aggregated by a `CouplingElement` with `CouplingElement.couplingType` equal to `CouplingElementEnum.switch` and `CouplingPort.couplingPortDetails.framePreemptionSupport` is set to true, then the attribute `CouplingPortFifo.trafficClassPreemptionSupport` shall be defined.⌋

[constr_3784] Applicable `CouplingPortFifo` as `predecessor` for `portScheduler` = `enhancedTrafficShaper`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPortScheduler` with `CouplingPortScheduler.portScheduler` equals to `EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper` shall only refer to `CouplingPortFifo` in the role `CouplingPortScheduler.predecessor` where the `CouplingPortFifo` has a `CouplingPortFifo.shaper` of kind `CouplingPortEnhancedTrafficShaper`.⌋

[constr_3785] Exclusive definition of `etsAvailableBandwidthInPercent` or `etsAvailableBandwidthInWeightValue`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPortEnhancedTrafficShaper` shall either define an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent` or an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue` value, but not both.⌋

[constr_3786] Consistent usage of either `etsAvailableBandwidthInPercent` or `etsAvailableBandwidthInWeightValue` for `portScheduler` = `enhancedTrafficShaper`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈All the `CouplingPortFifo` referenced by the same `CouplingPortScheduler` with `CouplingPortScheduler.portScheduler` equals to `EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper` (according to [constr_3784]) shall define in their `CouplingPortEnhancedTrafficShaper` the same kind of value. All shall use either an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent` or an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue` value.⌋

[constr_3787] Existence of `CouplingPortTrafficClassAssignment.trafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈Every `CouplingPortTrafficClassAssignment` shall have a `trafficClass` attribute defined.⌋

[constr_3788] Existence of `CouplingPortFifo.assignedTrafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingPortFifo`, exactly one value of attribute `CouplingPortFifo.assignedTrafficClass` shall exist.⌋

[constr_3789] Allowed values for `CouplingPortFifo.assignedTrafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The allowed values for `CouplingPortFifo.assignedTrafficClass` are 0..65535.⌋

[constr_3790] Existence of `CouplingPortDetails.defaultTrafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingPortDetails`, the attribute `CouplingPortDetails.defaultTrafficClass` shall exist.⌋

[constr_3791] Allowed values for `CouplingPortDetails.defaultTrafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The allowed values for `CouplingPortDetails.defaultTrafficClass` are 0..65535.⌋

[constr_3792] `FrameMapping` between identical bus systems

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈The `FrameTriggering`s referenced in the role `sourceFrame` and `targetFrame` shall be owned by `PhysicalChannel`s which in turn are owned by `CommunicationCluster`s of the same kind.⌋

[constr_3796] Applicable attributes of `IEEE1722TpIidcConnection` for category `IIDC`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈If the `category` of an `IEEE1722TpIidcConnection` is set to `IIDC`, then the collection of attributes shall be restricted to the following list:

- `iidcChannel`
- `iidcSy`
- `iidcTag`
- `iidcTCode`

⌋

[constr_3797] Applicable attributes of `IEEE1722TpIidcConnection` for category `IEC61883`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈If the `category` of an `IEEE1722TpIidcConnection` is set to `IEC61883`, then the collection of attributes shall be restricted to the following list:

- `iidcChannel`
- `iidcDataBlockSize`
- `iidcFractionNumber`
- `iidcSourcePacketHeader`
- `iidcStreamFormat`
- `iidcSy`
- `iidcTag`
- `iidcTCode`

⌋

[constr_3799] DDS Fire & Forget Method definition

Imposition time: CP: `IT_EcuExt`

⌈If a `DdsServiceInstanceOperationCp` represents a fire & forget method, then only the reference `DdsServiceInstanceOperationCp.operationRequestTriggering` shall exist and the reference in the role `DdsServiceInstanceOperationCp.operationResponseTriggering` shall not exist.⌋

[constr_3800] Existence of `DdsCpServiceInstance.discoveryType`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `discoveryType` shall exist.⌋

[constr_3801] Existence of `DdsCpServiceInstance.serviceInterfaceId`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `serviceInterfaceId` shall exist.⌋

[constr_3802] Existence of `DdsCpServiceInstance.majorVersion`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `AbstractServiceInstance.majorVersion` shall exist.⌋

[constr_3803] Existence of DdsCpServiceInstance.serviceMinorVersion

Imposition time: CP: IT_EcuExt

⌈For each DdsCpServiceInstance, the attribute in the role serviceMinorVersion shall exist.⌋

[constr_3804] Existence of DdsCpServiceInstance.transportProtocol

Imposition time: CP: IT_EcuExt

⌈For each DdsCpServiceInstance, the attribute in the role transportProtocol shall exist.⌋

[constr_3805] Existence of DdsServiceInstanceEventCp.eventTriggering

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceEventCp, the reference in the role eventTriggering shall exist.⌋

[constr_3806] Existence of DdsServiceInstanceOperationCp.operationRequestTriggering

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceOperationCp, the reference in the role operationRequestTriggering shall exist.⌋

[constr_3807] Existence of DdsServiceInstanceFieldCp reference

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceFieldCp, at least one of the references in the role

- notifier
- getter
- setter

shall exist.⌋

[constr_3808] Existence of DdsCpProvidedServiceInstance.providedServiceInstanceId

Imposition time: CP: IT_EcuExt

⌈For each DdsCpServiceInstance, the attribute in the role providedServiceInstanceId shall exist.⌋

[constr_3809] Existence of DdsCpConsumedServiceInstance.requiredServiceInstanceId

Imposition time: CP: IT_EcuExt

⌈For each DdsCpConsumedServiceInstance, the attribute in the role requiredServiceInstanceId shall exist.⌋

[constr_3810] Value range of DdsCpProvidedServiceInstance.providedServiceInstanceId

Imposition time: CP: IT_SysDesc

⌈The value of DdsCpProvidedServiceInstance.providedServiceInstanceId shall be in the range of 0..65534.⌋

[constr_3811] DdsCpProvidedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpProvidedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_3812] DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_3813] DdsCpProvidedServiceInstance unicast address references target

Imposition time: CP: IT_SysDesc

⌈Each of the following references:

- localUnicastAddress
- staticRemoteUnicastAddress

If specified, shall refer to an ApplicationEndpoint defining a UDP Port⌋

[constr_3814] Multiple SocketAddress entries with the same IP Address and Port in the context of a given EcuInstance

Imposition time: CP: IT_SysDesc

⌈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 and Port in the aggregated ApplicationEndpoint and NetworkEndpoint, (e.g., 192.168.1.1 and 7431, respectively) then:

- only one of these SocketAddress elements shall be referenced by DdsCpProvidedServiceInstances or DdsCpConsumedServiceInstances in the role localUnicastAddress.
- only one of these SocketAddress elements shall be referenced by DdsCpProvidedServiceInstances or DdsCpConsumedServiceInstances in the role staticRemoteUnicastAddress.

⌋

[constr_3815] DDS Service communication is restricted to one VLAN

Imposition time: CP: IT_SysDesc

⌈ApplicationEndpoints elements that are referenced by:

- localUnicastAddress and staticRemoteUnicastAddress shall belong to the same VLAN.

⌋

[constr_3816] Value range of DdsCpConsumedServiceInstance.requiredServiceInstanceId

Imposition time: CP: IT_SysDesc

⌈The value of DdsCpConsumedServiceInstance.requiredServiceInstanceId shall be in the range of 0..65534 or "ALL".⌋

[constr_3817] DdsCpConsumedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpConsumedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_3818] DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_3819] DdsCpConsumedServiceInstance unicast address references target

Imposition time: CP: IT_SysDesc

⌈

- localUnicastAddress
- staticRemoteUnicastAddress

If specified, shall refer to an ApplicationEndpoint defining a UDP Port.⌋

[constr_3820] DDS Service communication is restricted to one VLAN

Imposition time: CP: IT_SysDesc

⌈ApplicationEndpoints elements that are referenced by:

- localUnicastAddress and staticRemoteUnicastAddress shall belong to the same VLAN.

⌋

[constr_3821] DDS transformer configuration

Imposition time: CP: IT_SysDesc

⌈For each TransformationDescription variant that is a DdsTransformationDescription

- attribute protocol of TransformationTechnology shall be set to Dds
- attribute version of TransformationTechnology shall be set to 1.0.0
- attribute transformerClass of TransformationTechnology shall be set to serializer
- attribute hasInternalState of TransformationTechnology shall be set to false
- attribute needsOriginalData of TransformationTechnology shall be set to true
- attribute dataTransformationKind of DataTransformation shall be set to DataTransformationKindEnum.symmetric
- attribute executeDespiteDataUnavailability of DataTransformation shall be set to true
- attribute headerLength of BufferProperties shall be set to 0 (bits)

⌋

[constr_3822] No further transfomers in scope of the DDS transformer

Imposition time: CP: IT_SysDesc

⌈If a DataTransformation references a TransformationTechnology in the role transformerChain and that TransformationTechnology aggregates a DdsTransformationDescription in the role transformationDescription, then that DataTransformation shall not reference any other TransformationTechnology.⌋

[constr_3824] No explicitWakeupChannel reference for channelSynchronousWakeup = TRUE

Imposition time: IT_SysDesc

⌈If EcuInstance.channelSynchronousWakeup is set to TRUE for an EcuInstance, then all CommunicationConnectors aggregated in the role connector shall not define any CommunicationConnector.explicitWakeupChannel reference.⌋

[constr_3826] DdsCpTopic.ddsPartition reference restriction

Imposition time: CP: IT_EcuExt

⌈The DdsCpTopic that is aggregated by a DdsCpDomain in the role ddsTopic is only allowed to reference a DdsCpPartition in the role ddsPartition that is aggregated by the same DdsCpDomain.⌋

[constr_3827] Existence of `IkePqPresharedKey.pqPresharedKeyId`

Imposition time: CP: `IT_SysDesc`

⌈For each `IkePqPresharedKey`, the attribute `pqPresharedKeyId` shall exist.⌋

[constr_3828] Existence of `IkePqPresharedKey.pqPresharedKey`

Imposition time: CP: `IT_SysDesc`

⌈For each `IkePqPresharedKey`, the reference `pqPresharedKey` shall exist.⌋

[constr_3829] `CommunicationCluster.comMLightTimeout` not applicable for `EthernetCluster`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `CommunicationCluster.comMLightTimeout` shall not be used in `EthernetCluster`.⌋

[constr_4000] Local communication of mode switches

Imposition time: CP: `IT_EcuExt`

⌈Ports with `ModeSwitchInterface`s cannot be connected across ECU boundaries.⌋

[constr_5030] Uniqueness of `LinOrderedConfigurableFrame.index`

Imposition time: CP: `IT_SysDesc`

⌈`LinOrderedConfigurableFrame.index` shall always be set and be unique in the context of the aggregating `LinCommunicationConnector`.⌋

[constr_5031] Uniqueness of `FramePid.index`

Imposition time: CP: `IT_SysDesc`

⌈`FramePid.index` shall always be set and be unique in the context of the aggregating `AssignFrameIdRange`.⌋

[constr_5032] Maximal one `NmConfig` per `System` is allowed to be defined

Imposition time: CP: `IT_SysDesc`

⌈Each `System` element is allowed to reference at most one `NmConfig` element with the `fibexElement` reference.⌋

[constr_5049] Ethernet switch packet to traffic class assignment restriction

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈Every `CouplingPortTrafficClassAssignment` shall have at least one `priority` attribute defined.⌋

[constr_5050] VariableDataPrototype of COM Based Transformer

Imposition time: CP: `IT_SysDesc`

⌈The `VariableDataPrototype` of [TPS_SYST_02058] shall be typed by an `ApplicationRecordDataType` or an `ImplementationDataType` of `category` STRUCTURE.⌋

[constr_5051] Existence of CanFrameTriggering.identifier in case of bus mirror target

Imposition time: CP: IT_SysDesc

⌈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_5053] Existence of ISignalPort.handleInvalid

Imposition time: CP: IT_SysDesc

⌈If the ISignalPort has a networkRepresentationProps.invalidValue defined then the ISignalPort.communicationDirection shall equal in.⌋

[constr_5054] externalReplacement not applicable for ISignalPort.handleInvalid

Imposition time: CP: IT_SysDesc

⌈In the context of ISignalPort.handleInvalid the value externalReplacement shall not be used.⌋

[constr_5055] DataMapping of elements of PRPortPrototypes is not supported

Imposition time: CP: IT_EcuExt

⌈A DataMapping shall not map elements of PRPortPrototypes to SystemSignals⌋

[constr_5058] Value range for CryptoServiceQueue.queueSize

Imposition time: CP: IT_SysDesc

⌈If the CryptoServiceQueue.queueSize is defined it shall have a value which is equal or greater than 1.⌋

[constr_5060] Mapping of a SecuredIPdu into a LinFrame is not allowed

Imposition time: CP: IT_SysDesc

⌈The mapping of a SecuredIPdu into a LinFrame with a PduToFrameMapping is not allowed.⌋

[constr_5061] EthernetCommunicationConnectors and referencing SocketAddresses shall be in the same VLAN

Imposition time: CP: IT_SysDesc

⌈Each EthernetCommunicationConnector that is referenced by a SocketAddress in the role connector or multicastConnector shall be referenced by the same EthernetPhysicalChannel that aggregates the SoAdConfig that in turn aggregates the SocketAddress.⌋

[constr_5062] SOME/IP ProvidedServiceInstances of the same serviceInterface on one EcuInstance

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_5063] ProvidedServiceInstance.serviceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.serviceIdentifier is mandatory.⌋

[constr_5064] ProvidedServiceInstance.majorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.majorVersion is mandatory.⌋

[constr_5065] ProvidedServiceInstance.minorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.minorVersion is mandatory.⌋

[constr_5066] ProvidedServiceInstance.instanceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.instanceIdentifier is mandatory.⌋

[constr_5067] ProvidedServiceInstance shall be unique in respect of serviceIdentifier, instanceIdentifier, majorVersion

Imposition time: CP: IT_SysDesc

⌈On a VLAN each ProvidedServiceInstance shall have a different serviceIdentifier, instanceIdentifier and majorVersion value combination.⌋

[constr_5068] ProvidedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the ProvidedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_5069] ProvidedServiceInstance.remoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.remoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_5071] EventHandler.eventMulticastAddress reference target

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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).⌋

[constr_5073] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is aggregated by an EventHandler

Imposition time: CP: IT_SysDesc

⌈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 or activationUnicastAndMulticast that is aggregated by an EventHandler

Imposition time: CP: IT_SysDesc

⌈An EventHandler that aggregates a PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationMulticast or activationUnicastAndMulticast 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 or activationUnicastAndMulticast and allowed SoConIPduIdentifier references

Imposition time: CP: IT_SysDesc

⌈A PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast is allowed to reference SoConIPduIdentifiers only in the iPduIdentifierUdp role.⌋

[constr_5076] PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ProvidedServiceInstance

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈The ApplicationEndpoint that is referenced by an EventHandler in the role eventMulticastAddress 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_5081] ConsumedServiceInstance.serviceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.serviceIdentifier is mandatory.⌋

[constr_5082] ConsumedServiceInstance.majorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.majorVersion is mandatory.⌋

[constr_5083] ConsumedServiceInstance.minorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.minorVersion is mandatory.⌋

[constr_5084] ConsumedServiceInstance.instanceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.instanceIdentifier is mandatory.⌋

[constr_5085] ConsumedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the ConsumedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_5086] ConsumedServiceInstance.remoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.remoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_5087] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is referenced by a ConsumedEventGroup

Imposition time: CP: IT_SysDesc

⌈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.⌋

[constr_5088] PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ConsumedServiceInstance

Imposition time: CP: IT_SysDesc

⌈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`

Imposition time: CP: `IT_SysDesc`

⌈If the `PduActivationRoutingGroup` contains the `iPduIdentifierUdp` reference then the aggregating `ConsumedServiceInstance` shall contain a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP address.⌋

[constr_5090] `ApplicationEndpoint`s referenced by `ConsumedEventGroup`s and by the aggregating `ConsumedServiceInstance` shall be in the same VLAN

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_5091] Relevance of `tcpRole` attribute

Imposition time: CP: `IT_SysDesc`

⌈The attribute `tcpRole` shall only exist if the `StaticSocketConnection` is aggregated by a `SocketAddress` that defines a TCP Port in the aggregated `ApplicationEndpoint`.⌋

[constr_5092] Local and remoteAddress of a StaticSocketConnection shall define the same transport protocol

Imposition time: CP: `IT_SysDesc`

⌈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`.⌋

[constr_5093] `pncGatewayType` and `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈When multiple `CommunicationConnector`s 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`.⌋

[constr_5094] `pncGatewayType` and ECU

Imposition time: CP: `IT_SysDesc`

⌈When an ECU is connected to more than one `PhysicalChannel` and has a relation to a Partial Network then all `CommunicationConnector`s 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`).⌋

[constr_5095] Relationship between the timing behavior of the ConsumedEventGroup retry and the timing behavior of an Offer message

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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 0xffffff of the corresponding ProvidedServiceInstance.⌋

[constr_5097] DltLogChannel.txPduTriggering and DltLogChannel.rxPduTriggering shall point to GeneralPurposeIPdus of category DLT

Imposition time: CP: IT_SysDesc

⌈DltLogChannel shall only reference PduTriggerings that are pointing to GeneralPurposeIPdus of category DLT in the roles txPduTriggering and rxPduTriggering.⌋

[constr_5100] Compatibility of two MetaDataItemSets

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈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 ISignal-

IPdu 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 SenderReceiver-Interfaces is compatible (according to the definition of compatible specification of meta-data described in [constr_5100]).

⌋

[constr_5104] Assignment of a FlexrayFrame where allowDynamicLSdu-Length is set to true

Imposition time: CP: IT_SysDesc

⌈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

Imposition time: CP: IT_SysDesc

⌈Only the last Pdu in a FlexrayFrame is allowed to qualify according to [TPS_SYST_03085] to be of dynamic length.⌋

[constr_5106] ISignalGroup and ISignal referenced from ISignal-Triggering

Imposition time: CP: IT_SysDesc

⌈Either an ISignalGroup and all ISignals referenced from the ISignalGroup are also referenced from ISignalTriggerings aggregated at the same Physical-Channel or neither the ISignalGroup nor any of the ISignals referenced by the ISignalGroup shall be referenced from ISignalTriggerings.⌋

[constr_5110] Search for a collection of ServiceInstances is not supported

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.instanceIdentifier is not allowed to be set to the value ANY or ALL.⌋

[constr_5111] Existence of references TlvDataIdDefinition.tlvArgument, TlvDataIdDefinition.tlvRecordElement, and TlvDataIdDefinition.tlv-ImplementationDataTypeElement

Imposition time: CP: IT_SysDesc

⌈For each TlvDataIdDefinition, only one out of the following references shall exist:

- reference to ArgumentDataPrototype in the role tlvArgument

- reference to ApplicationRecordElement in the role tlvRecordElement
- reference to ImplementationDataTypeElement in the role tlv-ImplementationDataTypeElement.

⌋

[constr_5112] ImplementationDataType needs to be defined if a "new-world" variable-size ApplicationArrayDataType is mapped to a single SystemSignal

Imposition time: CP: IT_EcuExt

⌈A SenderReceiverInterface.dataElement that is typed by a "new-world" variable-size ApplicationArrayDataType according to [TPS_SWCT_01644] (see definition in Software Component Template [2]) is only allowed to be mapped to a single SystemSignal by the SenderReceiverToSignalMapping if a DataTypeMap exists that points to both the ApplicationArrayDataType and an ImplementationDataType that fulfills the conditions of a "new-world" dynamic size array data type according to [TPS_SWCT_01645] (see definition in Software Component Template [2]).⌋

[constr_5113] Mapping of "old-world" variable size arrays to a single SystemSignal is not supported.

Imposition time: CP: IT_EcuExt

⌈The SenderReceiverToSignalMapping is not allowed to map a dataElement that is typed by an "old-world" variable size array defined by [TPS_SWCT_01641] and [TPS_SWCT_01642] (see definition in Software Component Template [2]) to a single SystemSignal.⌋

[constr_5114] Semantics of InterpolationRoutine.isDefault

Imposition time: CP: IT_SysDesc

⌈For each SwRecordLayout that is referenced by one or more InterpolationRoutineMappings that are aggregated by InterpolationRoutineMappingSets that are referenced from a System in the role interpolationRoutineMappingSet, only one of the collection of aggregated InterpolationRoutines shall have attribute isDefault set to True.⌋

[constr_5116] Uniqueness of the symbols of software-components and BSW modules

Imposition time: CP: IT_EcuExt

⌈For all SwComponentPrototypes typed by an ApplicationSwComponentType, NvBlockSwComponentType or SensorActuatorSwComponentType mapped to a given EcuInstance by means of SwcToEcuMapping respectively SwcToApplicationPartitionMapping and ApplicationPartitionToEcuPartitionMapping the following restriction applies:

The symbolic name of an AtomicSwComponentType referenced by a respective SwComponentPrototype in the role type shall not overlap with the module implementation prefix (MIP) of any of the basic software-modules existing on the EcuInstance.

The symbolic name of an AtomicSwComponentType is derived from the value of

- AtomicSwComponentType.symbol, or if this attribute does not exist
- AtomicSwComponentType.shortName.

⌋

[constr_5117] Client-Server communication over Ethernet

Imposition time: CP: IT_SysDesc

⌈A SystemSignal that is referenced by a ClientServerToSignalMapping in the role callSignal or returnSignal shall only be referenced by an ISignal that in turn is referenced by an ISignalTriggering aggregated by an EthernetPhysicalChannel.⌋

[constr_5118] Value range of UdpProps.udpTtl

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of UdpProps.udpTtl shall be in the range of 1..255.⌋

[constr_5119] Value range of TcpProps.tcpTtl

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpTtl shall be in the range of 1..255.⌋

[constr_5120] Value range of TcpProps.tcpDelayedAckTimeout

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpDelayedAckTimeout shall be in the range of 0..0.5.⌋

[constr_5121] Value range of TcpProps.tcpSynMaxRtx

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpSynMaxRtx shall be in the range of 0..255.⌋

[constr_5122] Value range of TcpProps.tcpMaxRtx

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpMaxRtx shall be in the range of 0..255.⌋

[constr_5123] Value range of TcpProps.tcpKeepAliveProbesMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpKeepAliveProbesMax shall be in the range of 0..65535.⌋

[constr_5124] Value range of TcpProps.tcpReceiveWindowMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpProps.tcpReceiveWindowMax shall be in the range of 0..65535.⌋

[constr_5125] Value range of TcpIpIcmpv4Props.tcpIpIcmpV4Ttl

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of TcpIpIcmpv4Props.tcpIpIcmpV4Ttl shall be in the range of 1..255.⌋

[constr_5126] Value range of Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup shall be in the range of 0..255.⌋

[constr_5127] Value range of Ipv4FragmentationProps.tcpIpIpNumFragments

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4FragmentationProps.tcpIpIpNumFragments shall be in the range of 0..255.⌋

[constr_5128] Value range of Ipv4FragmentationProps.tcpIpIpNumReassDgrams

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4FragmentationProps.tcpIpIpNumReassDgrams shall be in the range of 0..65535.⌋

[constr_5129] Value range of Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount shall be in the range of 0..255.⌋

[constr_5130] Value range of Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize shall be in the range of 1500..65535.⌋

[constr_5131] Value range of Ipv6FragmentationProps.tcpIpIpReassembly-Timeout

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassembly-Timeout shall be in the range of 0.001..100.⌋

[constr_5132] Value range of Ipv6FragmentationProps.tcpIpIpReassembly-SegmentCount

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassembly-SegmentCount shall be in the range of 1..255.⌋

[constr_5133] Value range of Ipv6FragmentationProps.tcpIpIpTxFragment-BufferCount

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpTxFragment-BufferCount shall be in the range of 1..1000.⌋

[constr_5134] Value range of Ipv6FragmentationProps.tcpIpIpTxFragment-BufferSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpTxFragment-BufferSize shall be in the range of 1500..65535.⌋

[constr_5135] Value range of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin and Dhcpv6Props.tcpIpDhcpV6CnfDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin.⌋

[constr_5136] Value range of Dhcpv6Props.tcpIpDhcpV6InfDelayMin and Dhcpv6Props.tcpIpDhcpV6InfDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMin.⌋

[constr_5137] Value range of Dhcpv6Props.tcpIpDhcpV6SolDelayMin and Dhcpv6Props.tcpIpDhcpV6SolDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMin.⌋

[constr_5138] Value range of Ipv6NdpProps.tcpIpNdpSlaacDadNumberOf-Transmissions

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpSlaacDadNumberOf-Transmissions shall be in the range of 0..254.⌋

[constr_5139] Value range of Ipv6NdpProps.tcpIpNdpSlaacDad-RetransmissionDelay

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpSlaacDadRetransmission-Delay shall be in the range of 0..10.⌋

[constr_5140] Value range of Ipv6NdpProps.tcpIpNdpDefaultReachableTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDefaultReachableTime shall be in the range of 0..120.⌋

[constr_5141] Value range of Ipv6NdpProps.tcpIpNdpDefaultRetransTimer

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDefaultRetransTimer shall be in the range of 0..60.⌋

[constr_5142] Value range of Ipv6NdpProps.tcpIpNdpNumUnicast-Solicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations shall be in the range of 0..255.⌋

[constr_5143] Value range of Ipv6NdpProps.tcpIpNdpNumMulticast-Solicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpNumMulticastSolicitations shall be in the range of 0..255.⌋

[constr_5144] Value range of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue shall be in the range of 0..60.⌋

[constr_5145] Value range of Ipv6NdpProps.tcpIpNdpMinRandomFactor

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMinRandomFactor shall be in the range of 0..100.⌋

[constr_5146] Value range of Ipv6NdpProps.tcpIpNdpMaxRandomFactor

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRandomFactor shall be in the range of 0..100.⌋

[constr_5147] Value range of Ipv6NdpProps.tcpIpNdpDestinationCacheSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDestinationCacheSize shall be in the range of 1..254.⌋

[constr_5148] Value range of Ipv6NdpProps.tcpIpNdpPrefixListSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpPrefixListSize shall be in the range of 1..254.⌋

[constr_5149] Value range of Ipv6NdpProps.tcpIpNdpDefaultRouterListSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDefaultRouterListSize shall be in the range of 2..254.⌋

[constr_5151] Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations shall be in the range of 0..255.⌋

[constr_5152] Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitationDelay

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitationDelay shall be in the range of 0.001..60.⌋

[constr_5153] Value range of `Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval` shall be in the range of 0.001..60.⌋

[constr_5154] Value range of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit` shall be in the range of 1..255.⌋

[constr_5157] Mixing of Point-To-Point and Multi-Drop is not allowed in a `CouplingPortConnection`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The `CouplingPortConnection` is allowed to reference a `CouplingPort` either:

- in the role `firstPort` and/or `secondPort` or
- in the role `nodePort`

⌋

[constr_5158] Usage of `plcaProps` only allowed on 10BASE-T1S networks

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPort` is allowed to aggregate `plcaProps` only if:

- the `CouplingPort.physicalLayerType` is set to 10BASE-T1S
- the `CouplingPort.macLayerType` is set to `xMII`
- the `CouplingPort` is referenced by a `CouplingPortConnection` with the `nodePort` reference.

⌋

[constr_5159] Mandatory `CouplingPortConnection` settings if multi-drop feature is used

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingPortConnection` uses the `nodePort` reference then the attribute `CouplingPortConnection.plcaLocalNodeCount` and the attribute `CouplingPortConnection.plcaTransmitOpportunityTimer` shall be set to a value.⌋

[constr_5160] Mandatory `PlcaProps` settings if multi-drop feature is used

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingPort` is referenced by a `CouplingPortConnection` in the role `nodePort` then the `CouplingPort` shall aggregate the `PlcaProps` and the following attributes shall be set to a value:

- plcaMaxBurstCount
- plcaMaxBurstTimer
- plcaLocalNodeId

⌋

[constr_5162] Valid TextTableMapping in the context of SenderRecRecordElementMapping

Imposition time: CP: IT_EcuExt

⌈The aggregation of a TextTableMapping at SenderRecRecordElementMapping is only valid if the SenderRecRecordElementMapping also references a SystemSignal in the role systemSignal.⌋

[constr_5163] Existence of attribute IPSecRule.headerType

Imposition time: CP: IT_SysDesc

⌈For each IPSecRule, the attribute headerType shall exist.⌋

[constr_5164] Existence of attribute IPSecRule.ipProtocol

Imposition time: CP: IT_SysDesc

⌈For each IPSecRule, the attribute ipProtocol shall exist.⌋

[constr_5165] Existence of attribute IPSecRule.policy

Imposition time: CP: IT_SysDesc

⌈For each IPSecRule, the attribute policy shall exist.⌋

[constr_5166] Existence of IPduMapping.pduMaxLength

Imposition time: CP: IT_SysDesc

⌈If several IPduMappings refer to the same PduTriggering in IPduMapping.sourceIPdu, then all of these IPduMappings shall provide either no IPduMapping.pduMaxLength value, or the same IPduMapping.pduMaxLength value.⌋

[constr_5167] pncGatewayType and ECU over the whole system

Imposition time: CP: IT_SysDesc

⌈Only one PNC Gateway ECU in the whole System shall exist that sets on all its CommunicationConnectors the pncGatewayType to active.⌋

[constr_5168] pncGatewayType passive and connected ECUs

Imposition time: CP: IT_SysDesc

⌈For all CommunicationConnectors with pncGatewayType set to passive belonging to one PNC Gateway ECU, all connected counterpart CommunicationConnectors, where pncGatewaypncGatewayTypeType is set to active shall be-

long to one ECU, if dynamicPncToChannelMappingEnabled is set to TRUE for at least one of the affected CommunicationConnectors.⌋

[constr_5170] nmPassiveModeEnabled and dynamicPncToChannelMappingEnabled

Imposition time: CP: IT_SysDesc

⌈If nmPassiveModeEnabled is set to TRUE on a NmNode then dynamicPncToChannelMappingEnabled shall be set to FALSE on the according CommunicationConnector referring to the same CommunicationController.⌋

[constr_5175] RtePluginProps shall reference at least one EcucContainerValue representing a RteRipsPlugin

Imposition time: CP: IT_EcuExt

⌈If a FlatInstanceDescriptor owns are RtePluginProps this RtePluginProps shall define the associatedRtePlugin reference and/or the associatedCrossSwClusterComRtePlugin reference.⌋

[constr_5176] Existence of CpSoftwareCluster of category HOST_SOFTWARE_CLUSTER on one EcuInstance

Imposition time: CP: IT_SwCluSysDesc

⌈On each EcuInstance, exactly one CpSoftwareCluster of category HOST_SOFTWARE_CLUSTER shall exist.⌋

[constr_5177] Validity of reference CpSoftwareClusterToEcuInstanceMapping.swCluster

Imposition time: CP: IT_SwCluSysDesc

⌈A CpSoftwareClusterToEcuInstanceMapping that references a given CpSoftwareCluster in the role CpSoftwareClusterToEcuInstanceMapping.swCluster shall be aggregated by the same System (in the role System.mapping.swMapping) that also refers to the referenced CpSoftwareCluster in the role System.swCluster.⌋

[constr_5178] Existence of attribute CpSoftwareClusterResource.globalResourceId

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterResource, attribute globalResourceId shall exist.⌋

[constr_5179] Existence of attribute CpSoftwareClusterResource.isMandatory

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterResource, attribute isMandatory shall exist.⌋

[constr_5180] Allowed values for CpSoftwareClusterResource.globalResourceId

Imposition time: CP: IT_ResPool

⌈Attribute CpSoftwareClusterResource.globalResourceId shall not be set to 0.⌋

[constr_5181] Existence of attribute CpSoftwareClusterServiceResource.category

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterServiceResource, attribute category shall exist.⌋

[constr_5182] PRPortPrototypes are excluded as CpSoftwareCluster interfaces

Imposition time: CP: IT_SwCluSysDesc

⌈A CpSoftwareClusterCommunicationResource is not allowed to be mapped by a PortElementToCommunicationResourceMapping to an element of a PortInterface in the context of a PRPortPrototype.⌋

[constr_5183] PortElementToCommunicationResourceMapping shall reference exactly one element of a PortInterface

Imposition time: CP: IT_SwCluSysDesc

⌈For any given PortElementToCommunicationResourceMapping, either the reference

- parameterDataPrototype or
- modeDeclarationGroupPrototype or
- trigger or
- clientServerOperation or
- variableDataPrototype

shall exist.⌋

[constr_5184] CpSoftwareClusterServiceResource can be provided only once on an EcuInstance

Imposition time: CP: IT_SwCluSysDesc

⌈A CpSoftwareClusterServiceResource shall not be mapped by several CpSoftwareClusterToResourceMappings to CpSoftwareClusters in the provider role if the CpSoftwareClusters are mapped to the same EcuInstance by CpSoftwareClusterToEcuInstanceMappings.⌋

[constr_5185] Existence of attribute BinaryManifestProvideResource.globalResourceId

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute globalResourceId shall exist.⌋

[constr_5186] Existence of attribute BinaryManifestProvideResource.resourceGuardValue

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute resourceGuardValue shall exist.⌋

[constr_5187] Existence of attribute BinaryManifestProvideResource.supportsMultipleNotifierSets

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute supportsMultipleNotifierSets shall exist.⌋

[constr_5188] Existence of attribute BinaryManifestProvideResource.numberOfNotifierSets

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute numberOfNotifierSets shall exist.⌋

[constr_5189] Existence of reference BinaryManifestProvideResource.resourceDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, the reference in the role resourceDefinition shall exist.⌋

[constr_5190] Existence of aggregation BinaryManifestProvideResource.item

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, the aggregation in the role item shall exist at least once.⌋

[constr_5191] Consequence of attribute BinaryManifestProvideResource.item.category

Imposition time: CP: IT_BinObjMetaData

⌈The following values of attribute BinaryManifestProvideResource.item.category shall require the existence of aggregations:

- If category is set to PROVIDER_HANDLE and the attribute isUnused is not set to true then the aggregation BinaryManifestProvideResource.item.value shall exist
- If category is set to NOTIFIER_HANDLE and the attribute isUnused is not set to true then the aggregation BinaryManifestProvideResource.item.defaultValue shall exist
- If category is set to AUXILARY_ACTUAL_NUMBER_NOTIFIER_SETS then the aggregation BinaryManifestProvideResource.item.defaultValue shall exist

⌋

[constr_5192] Existence of attribute BinaryManifestRequireResource.globalResourceId

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, attribute globalResourceId shall exist.⌋

[constr_5193] Existence of attribute BinaryManifestRequireResource.resourceGuardValue

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, attribute resourceGuardValue shall exist.⌋

[constr_5194] Existence of reference BinaryManifestRequireResource.resourceDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, the reference in the role resourceDefinition shall exist.⌋

[constr_5195] Existence of aggregation BinaryManifestRequireResource.item

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, the aggregation in the role item shall exist at least once.⌋

[constr_5196] Consequence of attribute BinaryManifestRequireResource.item.category

Imposition time: CP: IT_BinObjMetaData

⌈The following values of attribute BinaryManifestRequireResource.item.category shall require the existence of aggregations:

- If category is set to PROVIDER_HANDLE then the aggregation BinaryManifestRequireResource.item.defaultValue shall exist

- If category is set to NOTIFIER_HANDLE then the aggregation BinaryManifestRequireResource.item.value shall exist

⌋

[constr_5197] Existence of aggregation BinaryManifestResourceDefinition.itemDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestResourceDefinition, the aggregation in the role itemDefinition shall exist at least once.⌋

[constr_5198] Allowed BinaryManifestResource.resourceDefinition

Imposition time: CP: IT_BinObjMetaData

⌈An BinaryManifestResourceDefinition shall only be referenced from a BinaryManifestResource that is aggregated in the same CpSoftwareClusterBinaryManifestDescriptor as the referenced BinaryManifestResourceDefinition.⌋

[constr_5199] Consequence of attribute BinaryManifestItem.auxiliaryField.category

Imposition time: CP: IT_BinObjMetaData

⌈If attribute BinaryManifestItem.auxiliaryField.category is set to value AUXILARY_CONNECTED_SW_CLUSTER_ID then attribute BinaryManifestItem.auxiliaryField.defaultValue shall exist.⌋

[constr_5200] Existence of attribute BinaryManifestItemDefinition.category

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemDefinition, attribute category shall exist.⌋

[constr_5201] Existence of attribute BinaryManifestItemDefinition.size

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemDefinition, attribute size shall exist.⌋

[constr_5202] Existence of attribute BinaryManifestItemNumericalValue.value

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemNumericalValue, attribute value shall exist.⌋

[constr_5203] Existence of attribute BinaryManifestItemPointerValue.symbol

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemPointerValue, attribute symbol shall exist.⌋

[constr_5204] Existence of attribute BinaryManifestMetaDataField.category

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute category shall exist.⌋

[constr_5205] Existence of attribute BinaryManifestMetaDataField.size

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute size shall exist.⌋

[constr_5206] Existence of attribute BinaryManifestMetaDataField.symbol

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute symbol shall exist .⌋

[constr_5207] Existence of attribute BinaryManifestMetaDataField.address

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute address shall exist.⌋

[constr_5208] Existence of System.swCluster

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.swCluster shall exist at least once.⌋

[constr_5209] Existence of reference CpSoftwareCluster.swComponentAssignment.swComponent

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.swCluster.swComponentAssignment.swComponent shall exist.⌋

[constr_5210] Existence of reference SystemMapping.portElementToComResourceMapping

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.portElementToComResourceMapping shall exist at least once.⌋

[constr_5211] Existence of reference PortElementToCommunicationResourceMapping.communicationResource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.portElementToComResourceMapping.communicationResource shall exist at least once.⌋

[constr_5212] Existence of reference SystemMapping.resourceToApplicationPartitionMapping

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping shall exist.⌋

[constr_5213] Existence of reference CpSoftwareClusterResourceToApplicationPartitionMapping.applicationPartition

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping.applicationPartition shall exist.⌋

[constr_5214] Existence of reference CpSoftwareClusterResourceToApplicationPartitionMapping.resource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping.resource shall exist.⌋

[constr_5215] Existence of reference CpSoftwareClusterToResourceMapping.serviceResource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.softwareClusterToResourceMapping.serviceResource shall exist.⌋

[constr_5216] Existence of reference CpSoftwareClusterToResourceMapping.requester and/or provider

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, at least one of the references System.mapping.softwareClusterToResourceMapping.requester or System.mapping.softwareClusterToResourceMapping.provider shall exist.⌋

[constr_5217] Existence of attribute BinaryManifestMetaDataField.value

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField of category IMMUTABLE_TABLES_CHECKSUM, attribute value shall exist.⌋

[constr_5218] Existence of attribute `BinaryManifestItemPointerValue.address`

Imposition time: CP: `IT_BinObjMetaData`

⌈For each `BinaryManifestItemPointerValue`, attribute `address` shall exist.⌋

[constr_5219] `CpSoftwareCluster` shall only be mapped to one `EcuInstance`

Imposition time: CP: `IT_SwCluSysDesc`

⌈Within the context of one `CpSoftwareCluster`, for all `CpSoftwareCluster.swComponentAssignment.swComponent` (and nested instances of `SwComponentPrototype`s) that are referenced by a `SwcToEcuMapping` in the role `component` the following condition shall be be fulfilled: all referencing `SwcToEcuMapping`s shall refer to the same `EcuInstance` in the role `ecuInstance` and this `EcuInstance` shall also be referenced in the role `ecuInstance` by all `CpSoftwareClusterToEcuInstanceMapping`s that also refer to said `CpSoftwareCluster` in the role `swCluster`.⌋

[constr_5220] Multiplicity of `EndToEndTransformationISignalProps.sourceId` in PROFILE_04m, PROFILE_07m, PROFILE_08m and PROFILE_44m

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_04m, PROFILE_07m, PROFILE_08m, or PROFILE_44m then the multiplicity of the `EndToEndTransformationISignalProps.sourceId` attribute shall be 1.⌋

[constr_5221] Multiplicity of `EndToEndTransformationISignalProps.sourceId` in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, and PROFILE_76

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, or PROFILE_76 then the multiplicity of the `EndToEndTransformationISignalProps.sourceId` attribute shall be 0.⌋

[constr_5222] Mandatory elements of `UdpNmCluster`

Imposition time: CP: `IT_SysDesc`

⌈The following attributes shall always be defined for the `UdpNmCluster`:

- `nmMsgCycleTime`
- `nmMessageTimeoutTime`
- `nmNetworkTimeout`
- `nmRemoteSleepIndicationTime`
- `nmRepeatMessageTime`

- nmWaitBusSleepTime
- communicationCluster

⌋

[constr_5223] Mandatory elements of UdpNmNode

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The following attributes shall always be defined for the UdpNmNode:

- nmMsgCycleOffset

⌋

[constr_5224] UdpNmNode.nmMsgCycleOffset < UdpNmCluster.nmMsgCycleTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmNode.nmMsgCycleOffset shall be smaller than the value of UdpNmCluster.nmMsgCycleTime.⌋

[constr_5225] UdpNmCluster.nmNetworkTimeout multiple of UdpNmCluster.nmMsgCycleTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmCluster.nmNetworkTimeout shall be n * UdpNmCluster.nmMsgCycleTime with n > 1.⌋

[constr_5226] UdpNmCluster.nmRepeatMessageTime multiple of UdpNmCluster.nmMsgCycleTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmCluster.nmRepeatMessageTime shall be n * UdpNmCluster.nmMsgCycleTime.⌋

[constr_5229] Existence of attribute E2EProfileCompatibilityProps.transitToInvalidExtended is mandatory for each EndToEndTransformationDescription

Imposition time: CP: IT_SysDesc

⌈For each EndToEndTransformationDescription, a reference to E2EProfileCompatibilityProps in the role e2eProfileCompatibilityProps shall exist and the referenced E2EProfileCompatibilityProps shall define a value for the attribute transitToInvalidExtended.⌋

[constr_5231] Allowed values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment

Imposition time: CP: IT_SysDesc

⌈The valid values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment shall be 8, 16, 32, 64, 128 or 256.⌋

[constr_5232] Triggering in case of application writing the selector field signal

Imposition time: CP: IT_SysDesc

⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or
  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then any ISignal other than the ISignal representing the selector field shall be mapped into that dynamic part alternative ISignalIPdu using the transferProperty set to pending.⌋

[constr_5233] Usage of invalidValue in case of application writing the selector field signal

Imposition time: CP: IT_SysDesc

⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or
  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then

- the ISignal representing the selector field shall either

- define no invalid value (ISignal.networkRepresentationProps.invalidValue) or
- the invalidValue defined shall be different than any of the defined selector field values for that MultiplexedIPdu.

⌋

[constr_5235] Maximum Frame.frameLength of the used bus protocol shall not be exceeded

Imposition time: CP: IT_SysDesc

⌈The Pdu.length used for an IPdu and the IPduMapping.pduMaxLength used for a targetIPdu shall not exceed the limitation of the maximum Frame.frameLength of the used bus protocol (e.g. CAN2.0 max. Frame.frameLength == 8Byte, CAN-FD Frame.frameLength == 64byte).⌋

[constr_5236] Restriction of IPduMapping.pduMaxLength

Imposition time: CP: IT_SysDesc

⌈IPduMapping.pduMaxLength shall be equal or greater than the maximum Pdu.length of sourceIPdu and targetIPdu. For a N:1 routing and 1:N routing, respectively, the maximum Pdu.length of all involved Pdus shall be used to evaluate a proper IPduMapping.pduMaxLength.⌋

[constr_5244] Value of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields

Imposition time: CP: IT_SysDesc

⌈If attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields is configured, then the value of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all variable-size arrays that are transported in the SOME/IP message.

In other words, for each variable-size array contained in the SOME/IP message, the numerical value of *maximum number of elements * sizeof(data type of array element)* shall be computed which yields the maximum number of bytes required to store the individual variable-size array.

The size of the attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields shall be set such that the highest value (or bigger) obtained from the individual computations for the contained variable-size arrays can fit into the length field. The unit of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields is bytes.⌋

[constr_5245] Value of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields

Imposition time: CP: IT_SysDesc

⌈If attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields is configured, then the value of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all strings that are transported in the SOME/IP message.

In other words, for each string contained in the SOME/IP message, the numerical value of *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding)* shall be computed which yields the maximum number of bytes required to store the individual string.

The size of the attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields shall be set such that the highest value (or bigger) obtained from the individual computations for the contained strings can fit into the length field. The unit of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields is bytes.⌋

[constr_5246] SOME/IP Transformation settings for strings in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈In the context of an ISignal the usage of DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField is only allowed if the SOMEIPTransformationISignalProps.sizeOfStringLengthFields is not defined.⌋

[constr_5247] Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField

Imposition time: CP: IT_SysDesc

⌈If the configuration of length field is done using DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField then the value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of elements * sizeof(data type of array element).*⌋

[constr_5248] Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField

Imposition time: CP: IT_SysDesc

⌈If the configuration of length field is done using DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField then the value of attribute DataPrototypeTransformationProps.transformationProps.size-

OfStringLengthField shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding).*⌋

[constr_5249] Existence of Pdu.length

Imposition time: CP: IT_SysDesc

⌈For each Pdu, the attribute length shall exist.⌋

[constr_5251] CouplingPort.connectionNegotiationBehavior shall exist

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute CouplingPort.connectionNegotiationBehavior shall be defined.⌋

[constr_5252] LinSlaveConfig.protocolVersion shall exist

Imposition time: CP: IT_SysDesc

⌈The attribute LinSlaveConfig.protocolVersion shall exist.⌋

[constr_5253] Value range of ISignal.length

Imposition time: CP: IT_SysDesc

⌈The value of ISignal.length shall be in the range of 0..34359738360 Bits.⌋

[constr_5254] Value range of MultiplexedIPdu.selectorFieldLength

Imposition time: CP: IT_SysDesc

⌈The value of MultiplexedIPdu.selectorFieldLength shall be in the range of 1..16 Bits.⌋

[constr_5258] TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping in case DataTransformation is used

Imposition time: CP: IT_SysDesc

⌈The ISignal that is referenced by a SystemSignal that in turn is referenced by a TriggerToSignalMapping in the role systemSignal shall have its length attribute set to the value of BufferProperties.headerLength attribute of the respective TransformationTechnology if the ISignal references a DataTransformation in the role dataTransformation that in turn references the TransformationTechnology.⌋

[constr_5259] PduTriggerings and FrameTriggerings of SecuredIPdu with useAsCryptographicIPdu = true

Imposition time: CP: IT_SysDesc

⌈In case that a SecuredIPdu is defined with useAsCryptographicIPdu = true as described by [TPS_SYST_02173] then:

- the PduTriggering of the AuthenticPdu
- the PduTriggering of the CryptographicPdu
- the FrameTriggering that references the Frame to which the AuthenticPdu is mapped
- the FrameTriggering that references the Frame to which the CryptographicPdu is mapped

shall be aggregated by the same PhysicalChannel.⌋

[constr_5262] SystemSignal used for Trigger communication shall not be part of any SystemSignalGroup

Imposition time: CP: IT_EcuExt

⌈A SystemSignal that is target of a TriggerToSignalMapping in the role systemSignal shall not be referenced by a SystemSignalGroup in the role systemSignal.⌋

[constr_5263] NetworkEndpoint.networkEndpointAddress restriction for IPv4

Imposition time: CP: IT_SysDesc

⌈A NetworkEndpoint shall not aggregate several Ipv4Configurations that have their ipv4AddressSource set to fixed.⌋

[constr_5264] NetworkEndpoint.networkEndpointAddress restriction for IPv6

Imposition time: CP: IT_SysDesc

⌈A NetworkEndpoint shall not aggregate several Ipv6Configurations that have their ipv6AddressSource set to fixed.⌋

[constr_5265] NetworkEndpoint.networkEndpointAddress restriction

Imposition time: CP: IT_SysDesc

⌈A NetworkEndpoint shall not aggregate an Ipv4Configuration and an Ipv6Configuration as networkEndpointAddress at the same time.⌋

[constr_5266] VariableDataPrototype of NvDataInterface shall not be mapped to a SystemSignal

Imposition time: CP: IT_EcuExt

⌈A VariableDataPrototype that is aggregated by a NvDataInterface shall not be referenced by

- SenderReceiverToSignalGroupMapping in the role dataElement and
- SenderReceiverToSignalMapping in the role dataElement.

⌋

[constr_5267] `VariableDataPrototype` of `NvDataInterface` shall not be mapped to a `SystemSignal` via a delegation to a `PortPrototype` with a `SenderReceiverInterface`

Imposition time: CP: `IT_EcuExt`

⌈If a `VariableDataPrototype` that is aggregated by a

- `SenderReceiverInterface` and that `SenderReceiverInterface` is referenced by a `PortPrototype` of a Composition and
- that `PortPrototype` is connected by a delegation connector with an inner `PortPrototype` of a `NvBlockSwComponentType` and
- that `PortPrototype` is typed by a `NvDataInterface`

then this `PortPrototype` shall not be referenced by:

- `SenderReceiverToSignalGroupMapping` in the role `dataElement` and
- `SenderReceiverToSignalMapping` in the role `dataElement`.

⌋

[constr_5268] Existence of `ContainedIPduProps.containedPduTriggering` reference

Imposition time: CP: `IT_SysDesc`

⌈If a `ContainedIPduProps` is aggregated at the `ContainerIPdu` in the role `ContainerIPdu.containedIPduTriggeringProps` then the reference `ContainedIPduProps.containedPduTriggering` shall exist.⌋

[constr_5271] Existence of attribute `BinaryManifestItem.isUnused`

Imposition time: CP: `IT_BinObjMetaData`

⌈For each `BinaryManifestItem`, the attribute `isUnused` shall exist.⌋

[constr_5272] Value of attribute `BinaryManifestItem.isUnused`

Imposition time: CP: `IT_BinObjMetaData`

⌈The attribute `BinaryManifestItem.isUnused` shall only permitted to be set to true if the related `BinaryManifestItemDefinition` has its attribute `isOptional` set to true.⌋

[constr_5273] One `ISignalTriggering` pair allowed per `EthernetPhysicalChannel` for a `ClientServerOperation`

Imposition time: CP: `IT_SysDesc`

⌈For each `EthernetPhysicalChannel` at most one pair of

- `ISignalTriggering` that refers to an `ISignal` that in turn refers to a `SystemSignal` that is referenced by a specific `ClientServerToSignalMapping` in the role `callSignal`

- ISignalTriggering that refers to an ISignal that in turn refers to a SystemSignal that is referenced by the same ClientServerToSignalMapping in the role returnSignal

shall exist, if the respective ISignalTriggerings are not referenced by a SomeipTpConnection in the role tpConcurrentProcessingSdu⌋

[constr_5274] ISignalTriggerings that represent the callSignal and returnSignal of the same ClientServerOperation on a PhysicalChannel shall be referenced by the same ClientServerToSignalMapping

Imposition time: CP: IT_SysDesc

⌈If on an EthernetPhysicalChannel an ISignalTriggering that refers to an ISignal that in turn refers to a SystemSignal that is referenced by a specific ClientServerToSignalMapping in the role callSignal is defined, then another ISignalTriggering shall be aggregated by the same EthernetPhysicalChannel and that ISignalTriggering shall refer to an ISignal that in turn refers to a SystemSignal that is referenced by the same ClientServerToSignalMapping in the role returnSignal, and vice versa.⌋

[constr_5306] Restriction of DltLogChannel.logChannelId attribute value

Imposition time: CP: IT_SysDesc

⌈The DltLogChannel.logChannelId attribute value shall be composed of maximum four ASCII characters.⌋

[constr_5307] Existence of DltLogChannel.logChannelId

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute logChannelId shall be defined.⌋

[constr_5308] Existence of DltLogChannel.nonVerboseMode

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute nonVerboseMode shall be defined.⌋

[constr_5309] Existence of DltConfig.sessionIdSupport

Imposition time: CP: IT_SysDesc

⌈For each DltConfig, the attribute sessionIdSupport shall be defined.⌋

[constr_5310] Existence of DltConfig.timestampSupport

Imposition time: CP: IT_SysDesc

⌈For each DltConfig, the attribute timestampSupport shall be defined.⌋

[constr_5311] Existence of DltLogChannel.logTraceDefaultLogThreshold

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute logTraceDefaultLogThreshold shall be defined.⌋

[constr_5312] Existence of DltLogChannel.defaultTraceState

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute defaultTraceState shall be defined.⌋

[constr_5313] Existence of DltLogChannel.txPduTriggering

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the reference to PduTriggering in the role txPduTriggering shall be defined.⌋

[constr_5314] DltLogChannel txPduTriggering and rxPduTriggering shall be on the same network

Imposition time: CP: IT_SysDesc

⌈The PduTriggerings that are referenced by a DltLogChannel in the role txPduTriggering and rxPduTriggering shall be aggregated by the same PhysicalChannel.⌋

[constr_5319] TCP endpoint using TLS_SERVER role can only serve provided service instances

Imposition time: CP: IT_SysDesc

⌈An ApplicationEndpoint that refers to TlsCryptoServiceMapping with category TLS_SERVER in the role tlsCryptoMapping is only allowed to be referenced by ProvidedServiceInstances in the role localUnicastAddress in case that the ProvidedServiceInstance does not have a remoteUnicastAddress defined.⌋

[constr_5320] TCP endpoint using TLS_CLIENT role can only serve consumed service instances

Imposition time: CP: IT_SysDesc

⌈An ApplicationEndpoint that refers to TlsCryptoServiceMapping with category TLS_CLIENT in the role tlsCryptoMapping is only allowed to be referenced by ConsumedServiceInstances in the role localUnicastAddress in case that the ConsumedServiceInstance does not have a remoteUnicastAddress defined.⌋

[constr_5321] Value range of Pdu.length

Imposition time: CP: IT_SysDesc

⌈The value of Pdu.length shall be in the range of 0..4294967295 Bytes.⌋

[constr_5322] Value range of ISignalToIPduMapping.startPosition

Imposition time: CP: IT_SysDesc

⌈The value of ISignalToIPduMapping.startPosition shall be in the range of 0..4294967295 Bits.⌋

[constr_5323] Value range of ISignalToIPduMapping.updateIndicationBitPosition

Imposition time: CP: IT_SysDesc

⌈The value of ISignalToIPduMapping.updateIndicationBitPosition shall be in the range of 0..4294967295 Bits.⌋

[constr_5326] Each local SocketAddress of an EcuInstance shall reference an EthernetCommunicationConnector in the role connector or multicastConnector

Imposition time: CP: IT_SysDesc

⌈If an EcuInstance uses a SocketAddress as local address, the SocketAddress shall refer to an EthernetCommunicationConnector of the EcuInstance, either via SocketAddress.connector if the SocketAddress represents a unicast address, or via SocketAddress.multicastConnector if the SocketAddress represents a multicast address.⌋

[constr_5327] Existence of attribute CpSoftwareCluster.category

Imposition time: CP: IT_SwCluSysDesc

⌈For each CpSoftwareCluster, attribute category shall exist.⌋

[constr_5328] Ecu Extract shall only contain outerPort DataMappings

Imposition time: CP: IT_EcuExt

⌈The System with category ECU_EXTRACT shall only contain DataMappings for VariableDataPrototypes, ClientServerOperations or Triggers that are referenced in the context of a PortPrototype of the SwComponentType that in turn is referenced by the RootSwCompositionPrototype.⌋

[constr_5329] SynchronousServerCallPoints for cross cluster communication are not supported

Imposition time: CP: IT_SwCluSysDesc

⌈A ClientServerOperation in the context of PortPrototype which is referenced by a PortElementToCommunicationResourceMapping in the role clientServerOperation is not allowed

- to be referenced by a SynchronousServerCallPoint.operation or
- to be connected to another ClientServerOperation in the context of a PortPrototype that in turn is referenced by SynchronousServerCallPoint.operation

⌋

[constr_5330] `ServiceInterface` elements shall belong to exactly one `ServiceInterface`

Imposition time: CP: `IT_SysDesc`

⌈If an element like

- a `VariableDataPrototype` that represents a `ServiceInterface` `event`
- a `ClientServerOperation` that represents a `ServiceInterface` `method`
- a `Collection` with `collectionSemantics` SO_SERVICE_FIELD that represents a `ServiceInterface` `field`
- a `Collection` with `collectionSemantics` SO_SERVICE_FIRE_AND_FORGET_METHOD that represents a “fire & forget” `method`

is referenced in the role `element` by a `Collection` that has the `collectionSemantics` set to SO_SERVICE_INTERFACE then this element shall not be referenced by any other `Collection` element that has the `collectionSemantics` SO_SERVICE_INTERFACE in the scope of the `System`.⌋

[constr_5331] No IP multicast in case of TCP

Imposition time: CP: `IT_SysDesc`

⌈The `ApplicationEndpoint` that is referenced in the role `eventMulticastAddress` from an `EventHandler` is only allowed to aggregate `UdpTp` in the role `tpConfiguration`.⌋

[constr_5334] Supported values for `CryptoServiceKey.length`

Imposition time: CP: `IT_SysDesc`

⌈The values defined for `CryptoServiceKey.length` shall be multiple of 8.⌋

[constr_5335] `CpSoftwareCluster.softwareClusterId` shall be unique in the scope of an `EcuInstance`

Imposition time: CP: `IT_SwCluSysDesc`

⌈The `softwareClusterId` shall be unique for each `CpSoftwareCluster` that is mapped to the same `EcuInstance` with the `CpSoftwareClusterToEcuInstanceMapping`.⌋

[constr_5336] Existence of `CpSoftwareCluster.softwareClusterId`

Imposition time: CP: `IT_SwCluSysDesc`

⌈For each `CpSoftwareCluster`, attribute `softwareClusterId` shall exist.⌋

[constr_5337] All CpSoftwareClusterToEcuInstanceMappings that are referencing the same EcuInstance shall define the same machineId

Imposition time: CP: IT_SysDesc

⌈All CpSoftwareClusterToEcuInstanceMappings that define a machineId and are referencing the same EcuInstance in the role ecuInstance shall have the same CpSoftwareClusterToEcuInstanceMapping.machineId value set.⌋

[constr_5344] Applicable transferProperty for GroupSignal and ISignalGroup

Imposition time: CP: IT_SysDesc

⌈

transferProperty on ISignalGroup	transferProperty on Group Signals	Semantic
• not set	Not set or pending for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the ISignalIPdu.
	Subset of Group Signals has transferProperty set to either triggered or triggeredWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group marks the ISignalIPdu for transmission.
	Subset of Group Signals has transferProperty set to either triggeredOnChange or triggered-OnChangeWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group and change of a Group Signal that has transferProperty set to triggeredOnChange or triggered-OnChangeWithoutRepetition causes immediate transmission of the ISignalIPdu.
• pending	Not set or pending for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the ISignalIPdu. See [TPS_SYST_02199].
• triggered • triggeredWithoutRepetition	Not set or pending for all Group Signals	Update of the Signal Group marks the ISignalIPdu for transmission. See [TPS_SYST_02199]
	Subset of Group Signals has transferProperty set to either triggered or triggeredWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group marks the ISignalIPdu for transmission. See [TPS_SYST_02200].
• triggeredOnChange • triggeredOnChangeWithout-Repetition	Not set or pending for all Group Signals	Update of Signal Group causes immediate transmission of the ISignalIPdu. See [TPS_SYST_02199]

▽

△

	Subset of Group Signals has transfer Property set to either `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` and the other Group Signals have `transferProperty` either not set or set to `pending`.	Update of Signal Group and change of a Group Signal that has `transferProperty` set to `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` causes immediate transmission of the `ISignalIPdu`. See [TPS_SYST_02200].

If the `ISignalToIPduMapping` refers to an `ISignalGroup` in the role `iSignalGroup` and the `ISignalIPdu` has an `EventControlledTiming` aggregated at the `TransmissionModeTiming` then combinations of `transferProperty` attribute settings for the `ISignalGroup` and the included `ISignal`s are supported as defined in this table.

⌋

[constr_5359] `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId`

Imposition time: CP: `IT_BinObjMetaData`

⌈The `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId` in case that the `softwareClusterId` is set in the `CpSoftwareCluster` that is referenced via `CpSoftwareClusterBinaryManifestDescriptor.cpSoftwareCluster`.⌋

[constr_5360] Cross cluster communication involving `NvBlockSwComponentType` is not supported

Imposition time: CP: `IT_SwCluSysDesc`

⌈A `PortElementToCommunicationResourceMapping` that is referencing a `CpSoftwareClusterCommunicationResource` in the role `communicationResource` is not allowed to reference:

- a `VariableDataPrototype` in the role `variableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `NvDataInterface` or
- a `VariableDataPrototype` in the role `variableDataPrototype` which is connected to another `VariableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` and typed by a `NvDataInterface` or
- a `ClientServerOperation` in the role `clientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface` or
- a `ClientServerOperation` in the role `clientServerOperation` which is connected to another `ClientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface`

⌋

[constr_5361] MACsec configuration is only allowed on CouplingPorts aggregated by switch ports and EthernetCommunicationController

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈Only if a CouplingPort is either aggregated by an EthernetCommunicationController or by a CouplingElement with couplingType set to switch, then this CouplingPort is allowed in turn to aggregate a MacSecProps in the role macSecProps.⌋

[constr_5362] Relation between ContainerIPdu and contained PduTriggerings on sender side

Imposition time: CP: IT_SysDesc

⌈In the scope of one EcuInstance, if a PduTriggering has a reference to an IPduPort where attribute communicationDirection is set to the value out, then that PduTriggering shall only be referenced at most once by ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering.⌋

[constr_5369] Consistency between SwcToApplicationPartitionMapping and CpSoftwareClusterToApplicationPartitionMapping

Imposition time: CP: IT_SwCluSysDesc

⌈If a CpSoftwareClusterToApplicationPartitionMapping exists between a CpSoftwareCluster and an ApplicationPartition, then all SwComponentPrototypes mapped to that ApplicationPartition by SwcToApplicationPartitionMapping shall be assigned to this CpSoftwareCluster (via CpSoftwareCluster.swComponentAssignment or CpSoftwareCluster.swComposition).⌋

[constr_5370] Restriction for SystemSignalToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignal

Imposition time: CP: IT_SwCluSysDesc

⌈If a DataMapping to a SystemSignal exists for the port element (e.g. VariableDataPrototype, ClientServerOperation) that is mapped by the PortElementToCommunicationResourceMapping to a CpSoftwareClusterCommunicationResource and a SystemSignalToCommunicationResourceMapping exists for the same SystemSignal then the SystemSignalToCommunicationResourceMapping shall map this SystemSignal to the same CpSoftwareClusterCommunicationResource.⌋

[constr_5371] Restriction for SystemSignalGroupToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignalGroup

Imposition time: CP: IT_SwCluSysDesc

⌈If a SenderReceiverToSignalGroupMapping to a SystemSignalGroup exists for the VariableDataPrototype that is mapped by the PortElementToCommunicationResourceMapping to a CpSoftwareClusterCommunication-

Resource and a SystemSignalGroupToCommunicationResourceMapping exists for the same SystemSignalGroup then the SystemSignalGroupToCommunicationResourceMapping shall map this SystemSignalGroup to the same CpSoftwareClusterCommunicationResource.⌋

[constr_5374] IPdu shall only be referenced once from a FlexrayTpConnection in the role directTpSdu or reversedTpSdu on a FlexrayCluster

Imposition time: CP: IT_SysDesc

⌈Each IPdu that is referenced in the role directTpSdu or reversedTpSdu from a FlexrayTpConnection that is aggregated by a FlexrayTpConfig that references a FlexrayCluster shall not be referenced a second time in the role directTpSdu or reversedTpSdu from any FlexrayTpConnection that is aggregated by a FlexrayTpConfig that references the same FlexrayCluster.⌋

[constr_5376] IPdu shall only be referenced once from a CanTpConnection in the role tpSdu on a CanCluster

Imposition time: CP: IT_SysDesc

⌈Each IPdu that is referenced in the role tpSdu from a CanTpConnection that is aggregated by a CanTpConfig that references a CanCluster shall not be referenced in the role tpSdu from a different CanTpConnection that is aggregated by a CanTpConfig that references the same CanCluster.⌋

[constr_5377] IPdu shall only be referenced once from a LinTpConnection in the role linTpNSdu on a LinCluster

Imposition time: CP: IT_SysDesc

⌈Each IPdu that is referenced in the role linTpNSdu from a LinTpConnection that is aggregated by a LinTpConfig that references a LinCluster shall not be referenced in the role linTpNSdu from a different LinTpConnection that is aggregated by a LinTpConfig that references the same LinCluster.⌋

[constr_5378] PduTriggering shall only be referenced once from a SomeipTpConnection in the role tpSdu or tpConcurrentProcessingSdu

Imposition time: CP: IT_SysDesc

⌈Each PduTriggering that is referenced in the role tpSdu or tpConcurrentProcessingSdu from a SomeipTpConnection shall not be referenced in the role tpSdu or tpConcurrentProcessingSdu from a different SomeipTpConnection.⌋

[constr_5379] IPdu shall only be referenced once from a J1939TpPg in the role sdu on a J1939Cluster

Imposition time: CP: IT_SysDesc

⌈Each IPdu that is referenced in the role sdu from a J1939TpPg that is aggregated by a J1939TpConfig that references a J1939Cluster shall not be referenced in the

role sdu from a different J1939TpPg that is aggregated by a J1939TpConfig that references the same J1939Cluster.⌋

[constr_5380] Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of one VLAN

Imposition time: CP: IT_SysDesc

⌈SoConIPduIdentifiers with the same headerId shall not be referenced by PduActivationRoutingGroups of different EventHandlers if

- one or several of these EventHandlers has the multicastThreshold set to a value>0 and
- all these EventHandlers are aggregated by ProvidedServiceInstances that reference ApplicationEndpoints with the localUnicastAddress reference that in turn are aggregated by SocketAddresses which contain a reference to the same EthernetCommunicationConnector in the role connector (i.e. the EventHandlers are located on the same VLAN).

except for the case that all these EventHandlers have the multicastThreshold set to the value 1.⌋

[constr_5382] Relation between the value of attributes offerCyclicDelay and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈In the context of any given SomeipSdServerServiceInstanceConfig, if the value of attribute offerCyclicDelay exists, it shall be less or equal to the value of attribute serviceOfferTimeToLive.⌋

[constr_5383] Relation between the value of attributes initialRepetitionsBaseDelay and initialRepetitionsMax and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈In the context of any given SomeipSdServerServiceInstanceConfig, if the value of attribute initialRepetitionsMax in initialOfferBehavior is greater than zero, the value of attribute serviceOfferTimeToLive shall be greater or equal to initialRepetitionsBaseDelay * 2^initialRepetitionsMax.⌋

[constr_5384] Existence of BusMirrorChannelMapping.mirroringProtocol

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorChannelMapping, the attribute mirroringProtocol shall exist.⌋

[constr_5385] Reception of UserData inside of a `NmPdu` by Applications is not supported

Imposition time: CP: `IT_SysDesc`

⌈A `SystemSignal` that is referenced by an `ISignal` that in turn is mapped via an `ISignalToIPduMapping` into a `NmPdu` shall not be mapped by a `DataMapping` that references a `RPortPrototype` with the `contextPort` reference in the `VariableDataPrototypeInSystemInstanceRef` that the `DataMapping` aggregates.⌋

[constr_5389] Dependency between `globalTimeTxPeriod` and `globalTimePortRole`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `EthGlobalTimeManagedCouplingPort.globalTimeTxPeriod` shall only be set to a value if the attribute `EthGlobalTimeManagedCouplingPort.globalTimePortRole` is set to `timeMaster` or `dynamic`.⌋

[constr_5390] The `globalTimePortRole` shall not be configured to `timeSlave` several times in the same `GlobalTimeDomain`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `globalTimePortRole` shall not be set to `timeSlave` for two or more `EthGlobalTimeManagedCouplingPort`s that are aggregated by the same `GlobalTimeDomain` (via `globalTimeDomainProperty`).⌋

[constr_5393] Existence of `clientId`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdDefinition`, the attribute `clientId` shall exist.⌋

[constr_5394] Existence of `clientServerOperation`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdDefinition`, the attribute `clientServerOperation` shall exist.⌋

[constr_5395] Existence of `physicalChannel`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈Each `CommunicationCluster` shall aggregate at least one `PhysicalChannel` in the role `physicalChannel`.⌋

[constr_5396] Existence of `ClientIdRange.lowerLimit`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdRange`, the attribute `lowerLimit` shall exist.⌋

[constr_5397] Existence of `ClientIdRange.upperLimit`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdRange`, the attribute `upperLimit` shall exist.⌋

[constr_5398] Existence of `CommunicationConnector.commController`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommunicationConnector`, the reference in the role `commController` shall exist.⌋

[constr_5399] Existence of `ecu`

Imposition time: CP: `IT_SysDesc`

⌈For each `ECUMapping`, the reference to `HwElement` in the role `ecu` shall exist.⌋

[constr_5400] Existence of `ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `ECUMapping`, the reference to `EcuInstance` in the role `ecuInstance` shall exist.⌋

[constr_5401] Existence of `communicationController`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommunicationControllerMapping`, the reference to `CommunicationController` in the role `communicationController` shall exist.⌋

[constr_5402] Existence of `hwCommunicationController`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommunicationControllerMapping`, the reference to `HwElement` in the role `hwCommunicationController` shall exist.⌋

[constr_5403] Existence of `communicationConnector`

Imposition time: CP: `IT_SysDesc`

⌈For each `HwPortMapping`, the reference to `CommunicationConnector` in the role `communicationConnector` shall exist.⌋

[constr_5404] Existence of `hwCommunicationPort`

Imposition time: CP: `IT_SysDesc`

⌈For each `HwPortMapping`, the reference to `HwPinGroup` in the role `hwCommunicationPort` shall exist.⌋

[constr_5405] Existence of `actionPointOffset`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `actionPointOffset` shall exist.⌋

[constr_5406] Existence of `bit`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `bit` shall exist.⌋

[constr_5407] Existence of `casRxLowMax`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster` the attribute `casRxLowMax` shall exist.⌋

[constr_5408] Existence of `coldStartAttempts`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `coldStartAttempts` shall exist.⌋

[constr_5409] Existence of `cycle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `cycle` shall exist.⌋

[constr_5410] Existence of `cycleCountMax`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `cycleCountMax` shall exist.⌋

[constr_5412] Existence of `dynamicSlotIdlePhase`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `dynamicSlotIdlePhase` shall exist.⌋

[constr_5414] Existence of `listenNoise`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `listenNoise` shall exist.⌋

[constr_5415] Existence of `macroPerCycle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `macroPerCycle` shall exist.⌋

[constr_5416] Existence of `macrotickDuration`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `macrotickDuration` shall exist.⌋

[constr_5417] Existence of `maxWithoutClockCorrectionFatal`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `maxWithoutClockCorrectionFatal` shall exist.⌋

[constr_5418] Existence of `maxWithoutClockCorrectionPassive`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `maxWithoutClockCorrection-Passive` shall exist.⌋

[constr_5419] Existence of `minislotActionPointOffset`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `minislotActionPointOffset` shall exist.⌋

[constr_5420] Existence of `minislotDuration`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `minislotDuration` shall exist.⌋

[constr_5421] Existence of `networkIdleTime`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `networkIdleTime` shall exist.⌋

[constr_5422] Existence of `networkManagementVectorLength`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `networkManagementVectorLength` shall exist.⌋

[constr_5423] Existence of `numberOfMinislots`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `numberOfMinislots` shall exist.⌋

[constr_5424] Existence of `numberOfStaticSlots`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `numberOfStaticSlots` shall exist.⌋

[constr_5425] Existence of `offsetCorrectionStart`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `offsetCorrectionStart` shall exist.⌋

[constr_5426] Existence of `payloadLengthStatic`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `payloadLengthStatic` shall exist.⌋

[constr_5428] Existence of `staticSlotDuration`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `staticSlotDuration` shall exist.⌋

[constr_5429] Existence of `symbolWindow`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `symbolWindow` shall exist.⌋

[constr_5431] Existence of `syncFrameIdCountMax`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `syncFrameIdCountMax` shall exist.⌋

[constr_5432] Existence of `transmissionStartSequenceDuration`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `transmissionStartSequenceDuration` shall exist.⌋

[constr_5433] Existence of `wakeupRxIdle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `wakeupRxIdle` shall exist.⌋

[constr_5434] Existence of `wakeupRxLow`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `wakeupRxLow` shall exist.⌋

[constr_5435] Existence of `wakeupRxWindow`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `wakeupRxWindow` shall exist.⌋

[constr_5436] Existence of `wakeupTxActive`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `wakeupTxActive` shall exist.⌋

[constr_5437] Existence of `wakeupTxIdle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `wakeupTxIdle` shall exist.⌋

[constr_5438] Existence of `sampleClockPeriod`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `sampleClockPeriod` shall exist.⌋

[constr_5439] Existence of `admitWithoutMessageId`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `admitWithoutMessageId` shall exist.⌋

[constr_5440] Existence of `baseCycle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `baseCycle` shall exist.⌋

[constr_5441] Existence of `cycleRepetition`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `cycleRepetition` shall exist.⌋

[constr_5442] Existence of `fifoDepth`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `fifoDepth` shall exist.⌋

[constr_5443] Existence of `msgIdMask`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `msgIdMask` shall exist.⌋

[constr_5444] Existence of `msgIdMatch`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `msgIdMatch` shall exist.⌋

[constr_5445] Existence of `fifoRange`

Imposition time: CP: `IT_SysDesc`

⌈Each `FlexrayFifoConfiguration` shall aggregate at least two `FlexrayFifoRange`s in the role `fifoRange`.⌋

[constr_5446] Existence of `rangeMax`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoRange`, the attribute `rangeMax` shall exist.⌋

[constr_5447] Existence of `rangeMin`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoRange`, the attribute `rangeMin` shall exist.⌋

[constr_5448] Existence of `channelName`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayPhysicalChannel`, the attribute `channelName` shall exist.⌋

[constr_5449] `LinCommunicationController.protocolVersion` shall exist

Imposition time: CP: `IT_SysDesc`

⌈The attribute `LinCommunicationController.protocolVersion` shall exist.⌋

[constr_5450] Existence of `index`

Imposition time: CP: `IT_SysDesc`

⌈For each `LinOrderedConfigurableFrame`, the attribute shall `index` shall exist.⌋

[constr_5451] Existence of `LinOrderedConfigurableFrame.frame` reference

Imposition time: CP: `IT_SysDesc`

⌈For each `LinOrderedConfigurableFrame`, the reference to `LinFrame` in the role `frame` shall exist.⌋

[constr_5452] Existence of `LinConfigurableFrame.frame` reference

Imposition time: CP: `IT_SysDesc`

⌈For each `LinConfigurableFrame`, the reference to `LinFrame` in the role `frame` shall exist.⌋

[constr_5453] Existence of `macMulticastAddress`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `MacMulticastGroup`, the attribute `macMulticastAddress` shall exist.⌋

[constr_5454] Existence of `vlanIdentifier`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanConfig`, the attribute `vlanIdentifier` shall exist.⌋

[constr_5455] Existence of `couplingType`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingElement`, the attribute `couplingType` shall exist.⌋

[constr_5456] Existence of `communicationCluster`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingElement`, the reference `communicationCluster` shall exist.⌋

[constr_5457] Existence of `defaultPriority`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanMembership`, the attribute `defaultPriority` shall exist.⌋

[constr_5458] Existence of `vlan`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanMembership`, the reference `vlan` shall exist.⌋

[constr_5462] Existence of `ingressPriority`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `EthernetPriorityRegeneration`, the attribute `ingressPriority` shall exist.⌋

[constr_5463] Existence of `regeneratedPriority`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `EthernetPriorityRegeneration`, the attribute `regeneratedPriority` shall exist.⌋

[constr_5464] Existence of `trafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingPortTrafficClassAssignment`, the attribute `trafficClass` shall exist.⌋

[constr_5466] Existence of `SenderReceiverToSignalMapping.dataElement`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderReceiverToSignalMapping`, the reference to `VariableDataPrototype` in the role `dataElement` shall exist.⌋

[constr_5467] Existence of `SenderReceiverToSignalMapping.systemSignal`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderReceiverToSignalMapping`, the reference to `SystemSignal` in the role `systemSignal` shall exist.⌋

[constr_5468] Existence of `SenderReceiverToSignalGroupMapping.dataElement`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderReceiverToSignalGroupMapping`, the reference to `VariableDataPrototype` in the role `dataElement` shall exist.⌋

[constr_5469] Existence of `SenderReceiverToSignalGroupMapping.signalGroup`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderReceiverToSignalGroupMapping`, the reference to `SystemSignalGroup` in the role `signalGroup` shall exist.⌋

[constr_5470] Existence of `SenderReceiverToSignalGroupMapping.typeMapping`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderReceiverToSignalGroupMapping`, the aggregation of `SenderRecCompositeTypeMapping` in the role `typeMapping` shall exist.⌋

[constr_5471] Existence of `SenderRecArrayElementMapping.indexedArrayElement`

Imposition time: CP: `IT_EcuExt`

⌈For each `SenderRecArrayElementMapping`, the aggregation in the role `indexedArrayElement` shall exist.⌋

[constr_5472] Existence of IndexedArrayElement.index

Imposition time: CP: IT_EcuExt

⌈For each IndexedArrayElement, the attribute index shall exist.⌋

[constr_5473] Existence of ClientServerToSignalMapping.callSignal

Imposition time: CP: IT_EcuExt

⌈For each ClientServerToSignalMapping, the reference to SystemSignal in the role callSignal shall exist.⌋

[constr_5474] Existence of ClientServerToSignalMapping.clientServer-Operation

Imposition time: CP: IT_EcuExt

⌈For each ClientServerToSignalMapping, the reference to ClientServer-Operation in the role clientServerOperation shall exist.⌋

[constr_5477] Existence of TriggerToSignalMapping.systemSignal

Imposition time: CP: IT_EcuExt

⌈For each TriggerToSignalMapping, the reference to SystemSignal in the role systemSignal shall exist.⌋

[constr_5478] Existence of TriggerToSignalMapping.trigger

Imposition time: CP: IT_EcuExt

⌈For each TriggerToSignalMapping, the reference to Trigger in the role trig-ger shall exist.⌋

[constr_5479] Existence of PncMapping.pncIdentifier

Imposition time: CP: IT_SysDesc

⌈For each PncMapping, the attribute pncIdentifier shall exist.⌋

[constr_5480] Existence of EcuResourceEstimation.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each EcuResourceEstimation, the reference to EcuInstance in the role ecuInstance shall exist.⌋

[constr_5481] Existence of SwcToSwcSignal.dataElement

Imposition time: CP: IT_SysDesc

⌈Each SwcToSwcSignal shall reference exactly two VariableDataPrototypes in the role dataElement.⌋

[constr_5482] Existence of SwcToSwcOperationArguments.direction

Imposition time: CP: IT_SysDesc

⌈For each SwcToSwcOperationArguments, the attribute direction shall exist.⌋

[constr_5483] Existence of SwcToSwcOperationArguments.operation

Imposition time: CP: IT_SysDesc

⌈Each SwcToSwcOperationArguments element shall reference exactly two ClientServerOperations in the role operation.⌋

[constr_5484] Existence of ForbiddenSignalPath.physicalChannel

Imposition time: CP: IT_SysDesc

⌈For each ForbiddenSignalPath, at least one reference to PhysicalChannel in the role physicalChannel shall exist.⌋

[constr_5485] Existence of PermissibleSignalPath.physicalChannel

Imposition time: CP: IT_SysDesc

⌈For each PermissibleSignalPath, at least one reference to PhysicalChannel in the role physicalChannel shall exist.⌋

[constr_5486] Existence of SwcToEcuMapping.component

Imposition time: CP: IT_SysDesc

⌈For each SwcToEcuMapping, the reference to SwComponentPrototype in the role component shall exist.⌋

[constr_5487] Existence of SwcToEcuMapping.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each SwcToEcuMapping the reference to EcuInstance in the role ecuInstance shall exist.⌋

[constr_5488] Existence of SwcToImplMapping.component

Imposition time: CP: IT_EcuExt

⌈For each SwcToImplMapping, the reference to SwComponentPrototype in the role component shall exist at least once.⌋

[constr_5489] Existence of SwcToImplMapping.componentImplementation

Imposition time: CP: IT_EcuExt

⌈For each SwcToImplMapping, the reference to SwcImplementation in the role componentImplementation shall exist.⌋

[constr_5491] Existence of ComponentClustering.clusteredComponent

Imposition time: CP: IT_SysDesc

⌈For each ComponentClustering, at least one reference to SwComponentPrototype in the role clusteredComponent shall exist.⌋

[constr_5492] Existence of ComponentSeparation.separatedComponent

Imposition time: CP: IT_SysDesc

⌈For each ComponentSeparation always two references to SwComponentPrototypes in the role separatedComponent shall exist.⌋

[constr_5493] Existence of J1939ControllerApplication.functionId

Imposition time: CP: IT_SysDesc

⌈For each J1939ControllerApplication, the attribute functionId shall exist.⌋

[constr_5494] Existence of BusMirrorChannel.busMirrorNetworkId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorChannel, the attribute busMirrorNetworkId shall exist.⌋

[constr_5495] Existence of BusMirrorCanIdRangeMapping.destinationBaseId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute destinationBaseId shall exist.⌋

[constr_5496] Existence of BusMirrorCanIdRangeMapping.sourceCanIdCode

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute sourceCanIdCode shall exist.⌋

[constr_5497] Existence of BusMirrorCanIdRangeMapping.sourceCanIdMask

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute sourceCanIdMask shall exist.⌋

[constr_5498] Existence of BusMirrorCanIdToCanIdMapping.remappedCanId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdToCanIdMapping, the attribute remappedCanId shall exist.⌋

[constr_5499] Existence of BusMirrorLinPidToCanIdMapping.remappedCanId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorLinPidToCanIdMapping, the attribute remappedCanId shall exist.⌋

[constr_9100] Existence of CanFrameTriggering.canAddressingMode

Imposition time: CP: IT_SysDesc

⌈For each CanFrameTriggering, the attribute canAddressingMode shall exist.⌋

[constr_9101] Existence of RxIdentifierRange.lowerCanId

Imposition time: CP: IT_SysDesc

⌈For each RxIdentifierRange, the attribute lowerCanId shall exist.⌋

[constr_9102] Existence of RxIdentifierRange.upperCanId

Imposition time: CP: IT_SysDesc

⌈For each RxIdentifierRange, the attribute upperCanId shall exist.⌋

[constr_9103] Existence of communicationDirection

Imposition time: CP: IT_SysDesc

⌈For each CommConnectorPort, the attribute communicationDirection shall exist.⌋

[constr_9105] Existence of DoIpTpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConfig, the aggregation of at least one DoIpTpConnection in the role tpConnection shall exist.⌋

[constr_9106] Existence of DoIpTpConnection.doIpSourceAddress

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role doIpSourceAddress shall exist.⌋

[constr_9107] Existence of DoIpTpConnection.doIpTargetAddress

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role doIpTargetAddress shall exist.⌋

[constr_9108] Existence of DoIpTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role tpSdu shall exist.⌋

[constr_9109] Existence of IPv6ExtHeaderFilterList.allowedIPv6ExtHeader

Imposition time: CP: IT_SysDesc

⌈For each IPv6ExtHeaderFilterList, the attribute allowedIPv6ExtHeader shall exist.⌋

[constr_9110] Existence of TcpOptionFilterList.allowedTcpOption

Imposition time: CP: IT_SysDesc

⌈For each TcpOptionFilterList, the attribute allowedTcpOption shall exist.⌋

[constr_9111] Existence of ApplicationEndpoint.networkEndpoint

Imposition time: CP: IT_SysDesc

⌈For each ApplicationEndpoint, the reference to NetworkEndpoint in the role networkEndpoint shall exist.⌋

[constr_9113] Existence of UdpTp.udpTpPort

Imposition time: CP: IT_SysDesc

⌈For each UdpTp, the aggregation of TpPort in the role udpTpPort shall exist.⌋

[constr_9114] Existence of TcpTp.tcpTpPort

Imposition time: CP: IT_SysDesc

⌈For each TcpTp, the aggregation of TpPort in the role tcpTpPort shall exist.⌋

[constr_9122] Existence of NetworkEndpoint.networkEndpointAddress

Imposition time: CP: IT_SysDesc

⌈For each NetworkEndpoint, the aggregation of NetworkEndpointAddress in the role networkEndpointAddress shall exist.⌋

[constr_9123] Existence of MacMulticastConfiguration.macMulticastGroup

Imposition time: CP: IT_SysDesc

⌈For each MacMulticastConfiguration, the reference to MacMulticastGroup in the role macMulticastGroup shall exist.⌋

[constr_9124] Existence of FlexrayFrameTriggering.allowDynamicLSduLength

Imposition time: CP: IT_SysDesc

⌈For each FlexrayFrameTriggering, the attribute allowDynamicLSduLength shall exist.⌋

[constr_9125] Existence of FlexrayFrameTriggering.payloadPreambleIndicator

Imposition time: CP: IT_SysDesc

⌈For each FlexrayFrameTriggering, the attribute payloadPreambleIndicator shall exist.⌋

[constr_9126] Existence of FlexrayAbsolutelyScheduledTiming.slotID

Imposition time: CP: IT_SysDesc

⌈For each FlexrayAbsolutelyScheduledTiming, the attribute slotID shall exist.⌋

[constr_9127] Existence of FlexrayAbsolutelyScheduledTiming.communicationCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayAbsolutelyScheduledTiming, the aggregation of CommunicationCycle in the role communicationCycle shall exist.⌋

[constr_9128] Existence of CycleCounter.CycleCounter

Imposition time: CP: IT_SysDesc

⌈For each CycleCounter, the attribute CycleCounter shall exist.⌋

[constr_9129] Existence of CycleRepetition.BaseCycle

Imposition time: CP: IT_SysDesc

⌈For each CycleRepetition, the attribute BaseCycle shall exist.⌋

[constr_9130] Existence of CycleRepetition.CycleRepetition

Imposition time: CP: IT_SysDesc

⌈For each CycleRepetition the attribute CycleRepetition shall exist.⌋

[constr_9131] Existence of FrameTriggering.frame

Imposition time: CP: IT_SysDesc

⌈For each FrameTriggering, the reference to Frame in the role frame shall exist.⌋

[constr_9132] Existence of LinSporadicFrame.substitutedFrame

Imposition time: CP: IT_SysDesc

⌈For each LinSporadicFrame, at least one reference to LinUnconditionalFrame in the role substitutedFrame shall exist.⌋

[constr_9133] Existence of LinEventTriggeredFrame.linUnconditionalFrame

Imposition time: CP: IT_SysDesc

⌈For each LinEventTriggeredFrame, at least one reference to LinUnconditionalFrame in the role linUnconditionalFrame shall exist.⌋

[constr_9134] Existence of ScheduleTableEntry.delay

Imposition time: CP: IT_SysDesc

⌈For each ScheduleTableEntry the attribute delay shall exist.⌋

[constr_9135] Existence of ScheduleTableEntry.positionInTable

Imposition time: CP: IT_SysDesc

⌈For each ScheduleTableEntry, the attribute positionInTable shall exist.⌋

[constr_9136] Existence of ApplicationEntry.frameTriggering

Imposition time: CP: IT_SysDesc

⌈For each ApplicationEntry, the reference to FrameTriggering in the role frameTriggering shall exist.⌋

[constr_9137] Existence of AssignFrameId.assignedFrameTriggering

Imposition time: CP: IT_SysDesc

⌈For each AssignFrameId, the reference to LinFrameTriggering in the role assignedFrameTriggering shall exist.⌋

[constr_9138] Existence of UnassignFrameId.unassignedFrameTriggering

Imposition time: CP: IT_SysDesc

⌈For each UnassignFrameId, the reference to LinFrameTriggering in the role unassignedFrameTriggering shall exist.⌋

[constr_9139] Existence of AssignFrameIdRange.startIndex

Imposition time: CP: IT_SysDesc

⌈For each AssignFrameIdRange, the attribute startIndex shall exist.⌋

[constr_9140] Existence of FramePid.index

Imposition time: CP: IT_SysDesc

⌈For each FramePid, the attribute index shall exist.⌋

[constr_9141] Existence of FramePid.pid

Imposition time: CP: IT_SysDesc

⌈For each FramePid, the attribute pid shall exist.⌋

[constr_9142] Existence of AssignNad.newNad

Imposition time: CP: IT_SysDesc

⌈For each AssignNad, the attribute newNad shall exist.⌋

[constr_9143] Existence of ConditionalChangeNad.byte

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute byte shall exist.⌋

[constr_9144] Existence of ConditionalChangeNad.id

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute id shall exist.⌋

[constr_9145] Existence of ConditionalChangeNad.invert

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute invert shall exist.⌋

[constr_9146] Existence of ConditionalChangeNad.mask

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute mask shall exist.⌋

[constr_9147] Existence of ConditionalChangeNad.newNad

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute newNad shall exist.⌋

[constr_9148] Existence of DataDumpEntry.byteValue

Imposition time: CP: IT_SysDesc

⌈For each DataDumpEntry, 5 byteValues shall be defined.⌋

[constr_9149] Existence of FreeFormat.byteValue

Imposition time: CP: IT_SysDesc

⌈For each FreeFormat, 8 byteValues shall be defined.⌋

[constr_9150] Existence of NmEcu.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each NmEcu, the reference to EcuInstance in the role ecuInstance shall exist.⌋

[constr_9151] Existence of nmDataCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmDataCycle shall exist.⌋

[constr_9152] Existence of nmRemoteSleepIndicationTime

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRemoteSleepIndicationTime shall exist.⌋

[constr_9153] Existence of nmRepeatMessageTime

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRepeatMessageTime shall exist.⌋

[constr_9154] Existence of nmRepetitionCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRepetitionCycle shall exist.⌋

[constr_9155] Existence of nmVotingCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmVotingCycle shall exist.⌋

[constr_9156] Existence of nmScheduleVariant

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmClusterCoupling, the attribute nmScheduleVariant shall exist.⌋

[constr_9157] Existence of nmBusloadReductionActive

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmBusloadReductionActive shall exist.⌋

[constr_9158] Existence of nmImmediateNmTransmissions

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmImmediateNmTransmissions shall exist.⌋

[constr_9159] Existence of nmMessageTimeoutTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmMessageTimeoutTime shall exist.⌋

[constr_9160] Existence of nmMsgCycleTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster the attribute nmMsgCycleTime shall exist.⌋

[constr_9161] Existence of nmNetworkTimeout

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmNetworkTimeout shall exist.⌋

[constr_9162] Existence of nmRemoteSleepIndicationTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmRemoteSleepIndicationTime shall exist.⌋

[constr_9163] Existence of nmRepeatMessageTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmRepeatMessageTime shall exist.⌋

[constr_9164] Existence of nmWaitBusSleepTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmWaitBusSleepTime shall exist.⌋

[constr_9165] Existence of nmBusloadReductionEnabled

Imposition time: CP: IT_SysDesc

⌈For each CanNmClusterCoupling, the attribute nmBusloadReductionEnabled shall exist.⌋

[constr_9166] Existence of nmImmediateRestartEnabled

Imposition time: CP: IT_SysDesc

⌈For each CanNmClusterCoupling, the attribute nmImmediateRestartEnabled shall exist.⌋

[constr_9167] Existence of J1939NodeName.arbitraryAddressCapable

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute arbitraryAddressCapable shall exist.⌋

[constr_9168] Existence of J1939NodeName.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute ecuInstance shall exist.⌋

[constr_9169] Existence of J1939NodeName.function

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute function shall exist.⌋

[constr_9170] Existence of J1939NodeName.functionInstance

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute functionInstance shall exist.⌋

[constr_9171] Existence of J1939NodeName.identitiyNumber

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute identitiyNumber shall exist.⌋

[constr_9172] Existence of J1939NodeName.industryGroup

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute industryGroup shall exist.⌋

[constr_9173] Existence of J1939NodeName.manufacturerCode

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute manufacturerCode shall exist.⌋

[constr_9174] Existence of J1939NodeName.vehicleSystem

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute vehicleSystem shall exist.⌋

[constr_9175] Existence of J1939NodeName.vehicleSystemInstance

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute vehicleSystemInstance shall exist.⌋

[constr_9176] Existence of StaticPart.iPdu

Imposition time: CP: IT_SysDesc

⌈For each StaticPart, the reference to ISignalIPdu in role iPdu shall exist.⌋

[constr_9177] Existence of DynamicPartAlternative.initialDynamicPart

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative, the attribute initialDynamicPart shall exist.⌋

[constr_9178] Existence of DynamicPartAlternative.initialDynamicPart

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative the attribute initialDynamicPart shall exist.⌋

[constr_9179] Existence of DynamicPartAlternative.iPdu

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative, the reference to ISignalIPdu in role iPdu shall exist.⌋

[constr_9180] Existence of DynamicPartAlternative.selectorFieldCode

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative, the attribute selectorFieldCode shall exist.⌋

[constr_9181] Existence of MultiplexedPart.segmentPosition

Imposition time: CP: IT_SysDesc

⌈For each MultiplexedPart the aggregation of SegmentPosition in role segmentPosition shall exist.⌋

[constr_9182] Existence of SegmentPosition.segmentByteOrder

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentByteOrder shall exist.⌋

[constr_9183] Existence of SegmentPosition.segmentLength

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentLength shall exist.⌋

[constr_9184] Existence of SegmentPosition.segmentPosition

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentPosition shall exist.⌋

[constr_9185] Existence of TransmissionModeCondition.dataFilter

Imposition time: CP: IT_SysDesc

⌈For each TransmissionModeCondition, the aggregation of DataFilter in the role dataFilter shall exist.⌋

[constr_9186] Existence of TransmissionModeCondition.iSignalInIPdu

Imposition time: CP: IT_SysDesc

⌈For each TransmissionModeCondition, the reference to ISignalToIPduMapping in the role iSignalInIPdu shall exist.⌋

[constr_9187] Existence of ModeDrivenTransmissionModeCondition.modeDeclaration

Imposition time: CP: IT_SysDesc

⌈For each ModeDrivenTransmissionModeCondition, the reference to ModeDeclaration in the role modeDeclaration shall exist.⌋

[constr_9188] Existence of ModeDrivenTransmissionModeCondition.timePeriod

Imposition time: CP: IT_SysDesc

⌈For each CyclicTiming, the aggregation of TimeRangeType in the role timePeriod shall exist.⌋

[constr_9189] Existence of EventControlledTiming.numberOfRepetitions

Imposition time: CP: IT_SysDesc

⌈For each EventControlledTiming, the attribute numberOfRepetitions shall exist.⌋

[constr_9190] Existence of TimeRangeType.value

Imposition time: CP: IT_SysDesc

⌈For each TimeRangeType, the attribute value shall exist.⌋

[constr_9191] Existence of RelativeTolerance.relative

Imposition time: CP: IT_SysDesc

⌈For each RelativeTolerance, the attribute relative shall exist.⌋

[constr_9192] Existence of AbsoluteTolerance.absolute

Imposition time: CP: IT_SysDesc

⌈For each AbsoluteTolerance, the attribute absolute shall exist.⌋

[constr_9193] Existence of `TriggerIPduSendCondition.modeDeclaration`

Imposition time: CP: `IT_SysDesc`

⌈For each `TriggerIPduSendCondition`, the reference to `ModeDeclaration` in role `modeDeclaration` shall exist.⌋

[constr_9194] Existence of `DcmIPdu.diagPduType`

Imposition time: CP: `IT_SysDesc`

⌈For each `DcmIPdu`, the attribute `diagPduType` shall exist.⌋

[constr_9195] Existence of `PduToFrameMapping.packingByteOrder`

Imposition time: CP: `IT_SysDesc`

⌈For each `PduToFrameMapping`, the attribute `packingByteOrder` shall exist.⌋

[constr_9196] Existence of `PduToFrameMapping.startPosition`

Imposition time: CP: `IT_SysDesc`

⌈For each `PduToFrameMapping`, the attribute `startPosition` shall exist.⌋

[constr_9197] Existence of `PduToFrameMapping.pdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `PduToFrameMapping`, the reference to `Pdu` in the role `pdu` shall exist.⌋

[constr_9198] Existence of `PduTriggering.iPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `PduTriggering`, the reference to `Pdu` in the role `iPdu` shall exist.⌋

[constr_9199] Existence of `ISignalIPduGroup.communicationDirection`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignalIPduGroup`, the attribute `communicationDirection` shall exist.⌋

[constr_9200] Existence of `ContainerIPdu.headerType`

Imposition time: CP: `IT_SysDesc`

⌈For each `ContainerIPdu` the attribute `headerType` shall exist.⌋

[constr_9201] Existence of `ContainerIPdu.rxAcceptContainedIPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `ContainerIPdu` the attribute `rxAcceptContainedIPdu` shall exist.⌋

[constr_9202] Existence of `ContainedIPduProps.collectionSemantics`

Imposition time: CP: `IT_SysDesc`

⌈For each `ContainedIPduProps` the attribute `collectionSemantics` shall exist.⌋

[constr_9203] Existence of SecuredIPdu.payload

Imposition time: CP: IT_SysDesc

⌈For each SecuredIPdu, the reference to PduTriggering in the role payload shall exist.⌋

[constr_9204] Existence of SecuredIPdu.secureCommunicationProps

Imposition time: CP: IT_SysDesc

⌈For each SecuredIPdu the aggregation of SecureCommunicationProps in the role secureCommunicationProps, shall exist.⌋

[constr_9205] Existence of SecureCommunicationProps.dataId

Imposition time: CP: IT_SysDesc

⌈For each SecureCommunicationProps, the attribute dataId shall exist.⌋

[constr_9206] Existence of CryptoServiceKey.length

Imposition time: CP: IT_SysDesc

⌈For each CryptoServiceKey, the attribute length shall exist.⌋

[constr_9210] Existence of InitialSdDelayConfig.initialDelayMaxValue aggregated by SomeipSdClientServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdClient-ServiceInstanceConfig in the role initialFindBehavior, the attribute initialDelayMaxValue shall exist.⌋

[constr_9211] Existence of InitialSdDelayConfig.initialDelayMinValue aggregated by SomeipSdClientServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdClient-ServiceInstanceConfig in the role initialFindBehavior, the attribute initialDelayMinValue shall exist.⌋

[constr_9212] Existence of SomeipSdClientEventGroupTimingConfig.time-ToLive

Imposition time: CP: IT_SysDesc

⌈For each SomeipSdClientEventGroupTimingConfig, the attribute timeTo-Live shall exist.⌋

[constr_9213] Existence of RequestResponseDelay.minValue aggregated by SomeipSdClientEventGroupTimingConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdClient-EventGroupTimingConfig in the role requestResponseDelay, the attribute minValue shall exist.⌋

[constr_9214] Existence of RequestResponseDelay.maxValue aggregated by SomeipSdClientEventGroupTimingConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdClient-EventGroupTimingConfig in the role requestResponseDelay, the attribute maxValue shall exist.⌋

[constr_9215] Existence of InitialSdDelayConfig.initialDelayMaxValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdServer-ServiceInstanceConfig in the role initialOfferBehavior, the attribute initialDelayMaxValue shall exist.⌋

[constr_9216] Existence of InitialSdDelayConfig.initialDelayMinValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdServer-ServiceInstanceConfig in the role initialOfferBehavior, the attribute initialDelayMinValue shall exist.⌋

[constr_9217] Existence of SomeipSdServerServiceInstance-Config.serviceOfferTimeToLive

Imposition time: CP: IT_SysDesc

⌈For each SomeipSdServerServiceInstanceConfig, the attribute service-OfferTimeToLive shall exist.⌋

[constr_9218] Existence of RequestResponseDelay.minValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdServer-ServiceInstanceConfig in the role requestResponseDelay, the attribute min-Value shall exist.⌋

[constr_9219] Existence of `RequestResponseDelay.maxValue` aggregated by `SomeipSdServerServiceInstanceConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `RequestResponseDelay` that is aggregated by a `SomeipSdServerServiceInstanceConfig` in the role `requestResponseDelay`, the attribute `maxValue` shall exist.⌋

[constr_9220] Existence of `RequestResponseDelay.minValue` aggregated by `SomeipSdServerEventGroupTimingConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `RequestResponseDelay` that is aggregated by a `SomeipSdServerEventGroupTimingConfig` in the role `requestResponseDelay`, the attribute `minValue` shall exist.⌋

[constr_9221] Existence of `RequestResponseDelay.maxValue` aggregated by `SomeipSdServerEventGroupTimingConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `RequestResponseDelay` that is aggregated by a `SomeipSdServerEventGroupTimingConfig` in the role `requestResponseDelay`, the attribute `maxValue` shall exist.⌋

[constr_9222] Existence of `ISignal.dataTypePolicy`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the attribute `dataTypePolicy` shall exist.⌋

[constr_9223] Existence of `ISignal.length`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the attribute `length` shall exist.⌋

[constr_9224] Existence of `ISignal.systemSignal`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the reference to `SystemSignal` in the role `systemSignal` shall exist.⌋

[constr_9225] Existence of `ISignalGroup.systemSignalGroup`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignalGroup`, the reference to `SystemSignalGroup` in the role `systemSignalGroup` shall exist.⌋

[constr_9226] Existence of `TpConfig.communicationCluster`

Imposition time: CP: `IT_SysDesc`

⌈For each `TpConfig`, the reference to `CommunicationCluster` in the role `communicationCluster` shall exist.⌋

[constr_9227] Existence of TpAddress.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each TpAddress, the attribute tpAddress shall exist.⌋

[constr_9228] Existence of FlexrayTpConfig.pduPool

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of FlexrayTpPduPool in the role pduPool shall exist at least once.⌋

[constr_9229] Existence of FlexrayTpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of TpAddress in the role tpAddress shall exist at least once.⌋

[constr_9230] Existence of FlexrayTpConfig.tpEcu

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of FlexrayTpEcu in the role tpEcu shall exist at least once.⌋

[constr_9231] Existence of FlexrayTpConnection.directTpSdu

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to IPdu in the role directTpSdu shall exist.⌋

[constr_9233] Existence of FlexrayTpConnection.receiver

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpNode in the role receiver shall exist at least once.⌋

[constr_9234] Existence of FlexrayTpConnection.tpConnectionControl

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpConnection-Control in the role tpConnectionControl shall exist.⌋

[constr_9235] Existence of FlexrayTpConnection.transmitter

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpNode in the role transmitter shall exist.⌋

[constr_9236] Existence of FlexrayTpEcu.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpEcu, the reference to EcuInstance in the role ecuInstance shall exist.⌋

[constr_9237] Existence of FlexrayTpEcu.fullDuplexEnabled

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpEcu, the attribute fullDuplexEnabled shall exist.⌋

[constr_9247] Existence of CanTpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpAddress in the role tpAddress shall exist at least once.⌋

[constr_9248] Existence of CanTpConfig.tpChannel

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpChannel in the role tpChannel shall exist at least once.⌋

[constr_9249] Existence of CanTpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpConnection in the role tpConnection shall exist at least once.⌋

[constr_9250] Existence of CanTpConfig.tpEcu

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpEcu in the role tpEcu shall exist at least once.⌋

[constr_9251] Existence of CanTpConfig.tpNode

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpNode in the role tpNode shall exist at least once.⌋

[constr_9252] Existence of CanTpConnection.addressingFormat

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the attribute addressingFormat shall exist.⌋

[constr_9253] Existence of CanTpConnection.canTpChannel

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the reference to CanTpChannel in the role canTpChannel shall exist.⌋

[constr_9254] Existence of CanTpConnection.dataPdu

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the reference to NPdu in the role dataPdu shall exist.⌋

[constr_9255] Existence of CanTpConnection.paddingActivation

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the attribute paddingActivation shall exist.⌋

[constr_9256] Existence of CanTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the reference to IPdu in the role tpSdu shall exist.⌋

[constr_9257] Existence of CanTpAddress.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each CanTpAddress, the attribute tpAddress shall exist.⌋

[constr_9258] Existence of CanTpEcu.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each CanTpEcu, the attribute ecuInstance shall exist.⌋

[constr_9259] Existence of LinTpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each LinTpConfig, at least one TpAddress shall be aggregated by LinTpConfig in the role tpAddress.⌋

[constr_9260] Existence of LinTpConnection.dataPdu

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, the reference to NPdu in the role dataPdu shall exist.⌋

[constr_9261] Existence of LinTpConnection.linTpNSdu

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, the reference to IPdu in the role linTpNSdu shall exist.⌋

[constr_9262] Existence of LinTpConnection.receiver

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, at least one reference to LinTpNode in the role receiver shall exist.⌋

[constr_9263] Existence of LinTpConnection.transmitter

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, the reference to LinTpNode in the role transmitter shall exist.⌋

[constr_9264] Existence of J1939TpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one TpAddress shall be aggregated in the role tpAddress.⌋

[constr_9265] Existence of J1939TpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one J1939TpConnection shall be aggregated in the role tpConnection.⌋

[constr_9266] Existence of J1939TpConfig.tpNode

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one J1939TpNode shall be aggregated in the role tpNode.⌋

[constr_9270] Existence of TlsCryptoCipherSuite.version

Imposition time: CP: IT_SysDesc

⌈For each TlsCryptoCipherSuite, the attribute version shall exist.⌋

[constr_9271] Existence of TlsPskIdentity.pskIdentity

Imposition time: CP: IT_SysDesc

⌈For each TlsPskIdentity, the attribute pskIdentity shall exist.⌋

[constr_9273] Existence of DataTransformation.executeDespiteDataUnavailability

Imposition time: CP: IT_SysDesc

⌈For each DataTransformation, the attribute executeDespiteDataUnavailability shall exist.⌋

[constr_9274] Existence of DataTransformation.transformerChain

Imposition time: CP: IT_SysDesc

⌈For each DataTransformation, at least one reference to TransformationTechnology in the role transformerChain shall exist.⌋

[constr_9275] Existence of TransformationTechnology.bufferProperties

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, a BufferProperties element shall be aggregated by TransformationTechnology in the role bufferProperties.⌋

[constr_9276] Existence of TransformationTechnology.protocol

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute protocol shall exist.⌋

[constr_9277] Existence of TransformationTechnology.transformerClass

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute transformerClass shall exist.⌋

[constr_9278] Existence of TransformationTechnology.version

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute version shall exist.⌋

[constr_9279] Existence of BufferProperties.headerLength

Imposition time: CP: IT_SysDesc

⌈For each BufferProperties, the attribute headerLength shall exist.⌋

[constr_9280] Existence of BufferProperties.inPlace

Imposition time: CP: IT_SysDesc

⌈For each BufferProperties, the attribute inPlace shall exist.⌋

[constr_9281] Existence of TransformationISignalProps.dataPdu

Imposition time: CP: IT_SysDesc

⌈For each TransformationISignalProps, the reference to TransformationTechnology in the role transformer shall exist.⌋

[constr_9282] Existence of SOMEIPTransformationDescription.alignment

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute alignment shall exist.⌋

[constr_9283] Existence of SOMEIPTransformationDescription.byteOrder

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute byteOrder shall exist.⌋

[constr_9284] Existence of SOMEIPTransformationDescription.interfaceVersion

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute interfaceVersion shall exist.⌋

[constr_9285] Existence of DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr

Imposition time: CP: IT_SysDesc

⌈For each DataPrototypeInSenderReceiverInterfaceInstanceRef, the reference to DataPrototype in the role targetDataPrototypeInSr shall exist.⌋

[constr_9286] Existence of DataPrototypeInClientServerInterface-InstanceRef.targetDataPrototypeInCs

Imposition time: CP: IT_SysDesc

⌈For each DataPrototypeInClientServerInterfaceInstanceRef, the reference to DataPrototype in the role targetDataPrototypeInCs shall exist.⌋

[constr_9287] Existence of EndToEndTransformationDescription.profile-Name

Imposition time: CP: IT_SysDesc

⌈For each EndToEndTransformationDescription, the attribute profileName shall exist.⌋

[constr_9288] Existence of TlvDataIdDefinition.id

Imposition time: CP: IT_SysDesc

⌈For each TlvDataIdDefinition, the attribute id shall exist.⌋

[constr_9289] Existence of FrameMapping.sourceFrame

Imposition time: CP: IT_SysDesc

⌈For each FrameMapping, the reference to FrameTriggering in the role source-Frame shall exist.⌋

[constr_9290] Existence of FrameMapping.targetFrame

Imposition time: CP: IT_SysDesc

⌈For each FrameMapping, the reference to FrameTriggering in the role target-Frame shall exist.⌋

[constr_9291] Existence of Gateway.ecu

Imposition time: CP: IT_SysDesc

⌈For each Gateway, the reference to EcuInstance in the role ecu shall exist.⌋

[constr_9292] Existence of IPduMapping.sourceIPdu

Imposition time: CP: IT_SysDesc

⌈For each IPduMapping, the reference to PduTriggering in the role sourceIPdu shall exist.⌋

[constr_9293] Existence of IPduMapping.targetIPdu

Imposition time: CP: IT_SysDesc

⌈Each IPduMapping shall aggregate a TargetIPduRef in the role targetIPdu.⌋

[constr_9294] Existence of TargetIPduRef.targetIPdu

Imposition time: CP: IT_SysDesc

⌈For each TargetIPduRef, the reference to PduTriggering in the role target-IPdu shall exist.⌋

[constr_9295] Existence of PduMappingDefaultValue.defaultValueElement

Imposition time: CP: IT_SysDesc

⌈For each PduMappingDefaultValue, at least one DefaultValueElement shall be aggregated by PduMappingDefaultValue in the role defaultValueElement.⌋

[constr_9296] Existence of DefaultValueElement.elementPosition

Imposition time: CP: IT_SysDesc

⌈For each DefaultValueElement, the attribute elementPosition shall exist.⌋

[constr_9297] Existence of DefaultValueElement.elementByteValue

Imposition time: CP: IT_SysDesc

⌈For each DefaultValueElement, the attribute elementByteValue shall exist.⌋

[constr_9298] Existence of ISignalMapping.sourceSignal

Imposition time: CP: IT_SysDesc

⌈For each ISignalMapping, the reference to ISignalTriggering in the role sourceSignal shall exist.⌋

[constr_9299] Existence of ISignalMapping.targetSignal

Imposition time: CP: IT_SysDesc

⌈For each ISignalMapping, the reference to ISignalTriggering in the role targetSignal shall exist.⌋

[constr_9300] Existence of FlatMap.instance

Imposition time: CP: IT_EcuExt

⌈For each FlatMap, at least one FlatInstanceDescriptor shall be aggregated by FlatMap in the role instance.⌋

[constr_9301] Existence of AliasNameAssignment.shortLabel

Imposition time: CP: IT_EcuExt

⌈For each AliasNameAssignment, the attribute shortLabel shall exist.⌋

[constr_9302] Existence of GlobalTimeDomain.domainId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a GlobalTimeDomain defines a GlobalTimeDomain.globalTimeMaster or GlobalTimeDomain.slave, then the attribute GlobalTimeDomain.domainId shall exist.⌋

[constr_9303] Existence of GlobalTimeMaster.communicationConnector

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the reference to CommunicationConnector in the role communicationConnector shall exist.⌋

[constr_9304] Existence of GlobalTimeMaster.isSystemWideGlobalTimeMaster

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the attribute isSystemWideGlobalTimeMaster shall exist.⌋

[constr_9305] Existence of GlobalTimeMaster.syncPeriod

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the attribute syncPeriod shall exist.⌋

[constr_9306] Existence of GlobalTimeSlave.communicationConnector

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeSlave, the reference to CommunicationConnector in the role communicationConnector shall exist.⌋

[constr_9307] Existence of GlobalTimeGateway.master

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeGateway, the reference to GlobalTimeMaster in the role master shall exist.⌋

[constr_9308] Existence of GlobalTimeGateway.slave

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeGateway, the reference to GlobalTimeSlave in the role slave shall exist.⌋

[constr_9310] Existence of GlobalTimeCanSlave.crcValidated

Imposition time: CP: IT_SysDesc

⌈For each GlobalTimeCanSlave, the attribute crcValidated shall exist.⌋

[constr_9311] Existence of EthGlobalTimeDomainProps.messageCompliance

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthGlobalTimeDomainProps, the attribute messageCompliance shall exist.⌋

[constr_9312] Existence of EthGlobalTimeManagedCouplingPort.pdelayResponseEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthGlobalTimeManagedCouplingPort, the attribute pdelayResponseEnabled shall exist.⌋

[constr_9313] Existence of GlobalTimeCouplingPortProps.propagationDelay

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeCouplingPortProps, the attribute propagationDelay shall exist.⌋

[constr_9314] Existence of GlobalTimeFrMaster.crcSecured

Imposition time: CP: IT_SysDesc

⌈For each GlobalTimeFrMaster, the attribute crcSecured shall exist.⌋

[constr_9315] Existence of GlobalTimeFrSlave.crcValidated

Imposition time: CP: IT_SysDesc

⌈For each GlobalTimeFrSlave, the attribute crcValidated shall exist.⌋

[constr_9317] StateDependentFirewall.firewallStateModeDeclaration reference restriction

Imposition time: CP: IT_SysDesc

⌈Each StateDependentFirewall shall only reference ModeDeclarations in the role firewallStateModeDeclaration that are aggregated by the same ModeDeclarationGroup.⌋

[constr_9318] Reception of CanFrameTriggerings with the same identifier by an EcuInstance

Imposition time: CP: IT_SysDesc

⌈For all CanFrameTriggerings on the same PhysicalChannel that refer to a FramePort with the communicationDirection = in of the same EcuInstance the condition applies that no two of these CanFrameTriggerings shall have the same identifier and the same canAddressingMode assigned.⌋

[constr_9319] Value of BusMirrorChannelMappingCan.mirroringProtocol

Imposition time: CP: IT_SysDesc

⌈Within the scope of a BusMirrorChannelMappingCan, if the (see [constr_3468]) PduTriggering referenced in the role BusMirrorChannelMappingCan.targetPduTriggering is in turn referenced in the role pduTriggering by a CanFrameTriggering where the aggregation in the role canXlFrameTriggeringProps exists, then the value of attribute mirroringProtocol shall only be set to MirroringProtocolEnum.version1.⌋

[constr_9320] Value of BusMirrorChannelMappingFlexray.mirroringProtocol

Imposition time: CP: IT_SysDesc

⌈The value of attribute BusMirrorChannelMappingFlexray.mirroringProtocol shall only be set to MirroringProtocolEnum.version1.⌋

[constr_9321] Same time base for all `BusMirrorChannelMapping`s of one `EcuInstance`

Imposition time: CP: `IT_SysDesc`

⌈All `BusMirrorChannelMapping`s that are referencing the same `EcuInstance` in the role `ecuInstance` shall reference the same `GlobalTimeDomain` in the role `globalTimeDomain`.⌋

[constr_9326] Exclusive existence of `ISignalTriggering.iSignal` and `ISignalTriggering.iSignalGroup`

Imposition time: CP: `IT_SysDesc`

⌈Each `ISignalTriggering` shall either define an `ISignalTriggering.iSignal` or an `ISignalTriggering.iSignalGroup` reference.⌋

[constr_9330] Derivation of network representation in case that several `DataMapping`s are defined that map the same `SystemSignal` to different `VariableDataPrototype`s

Imposition time: CP: `IT_SysDesc`

⌈If several `DataMapping`s are defined that map the same `SystemSignal` to different `VariableDataPrototype`s then

- all `ISignal`s that reference this `SystemSignal` shall define `networkRepresentationProps` or
- if `networkRepresentationProps` are not specified on the `ISignal` level (and are therefore derived from the `ImplementationDataType`) then the different `DataMapping`s shall reference `VariableDataPrototype`s that in turn reference the identical `ImplementationDataType`.

⌋

[constr_9331] E2E protection of a `ClientServerOperation`

Imposition time: CP: `IT_SysDesc`

⌈If an `ISignal` aggregates `EndToEndTransformationISignalProps` and references a `SystemSignal` that in turn is referenced by a `ClientServerToSignalMapping` in the role `callSignal`, then the `EndToEndTransformationISignalProps` settings and the `EndToEndTransformationDescription` settings defined in the `TransformationTechnology` that is referenced by the `EndToEndTransformationISignalProps` shall have the same values for the following attributes:

- `EndToEndTransformationDescription.profileName`
- `EndToEndTransformationDescription.offset`
- `EndToEndTransformationISignalProps.sourceId`

as for the `ISignal` that is referenced by the `SystemSignal` that in turn is referenced by the same `ClientServerToSignalMapping` in the role `returnSignal`.⌋

[constr_9332] Existence of J1939TpConnection.tpProtocolType

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConnection, the attribute tpProtocolType shall exist.⌋

[constr_9333] FibexElements in ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈Each FibexElement that is used in the ECU_EXTRACT shall be referenced by the System element in the role fibexElement.⌋

[constr_9343] Allowed J1939ProtectedIPdu.payload reference target

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈A J1939ProtectedIPdu is only allowed to reference a PduTriggering with the payload reference that in turn references an ISignalIPdu in the role iPdu to which an ISignalGroup is mapped that aggregates EndToEndTransformationISignalProps in the role transformationISignalProps which references an EndToEndTransformationDescription with profileName PROFILE_76.⌋

[constr_9346] Existence of EthernetVlanTranslationTable.translatedVlanId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthernetVlanTranslationTable, the attribute translatedVlanId shall exist.⌋

[constr_9347] Range of EthernetVlanTranslationTable.ingressVlanId and EthernetVlanTranslationTable.translatedVlanId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of ingressVlanId and translatedVlanId shall be in the range 0..4095.⌋

[constr_9348] EthernetVlanTranslationTable.translatedVlanId and vlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort defines an EthernetVlanTranslationTable via the CouplingPortDetails then the CouplingPort shall have a vlanMembership defined that references an EthernetPhysicalChannel that has the same vlanIdentifier value defined as the translatedVlanId value in the EthernetVlanTranslationTable.⌋

[constr_9349] An ISignal shall not be used for SOME/IP and DDS transport at the same time

Imposition time: CP: IT_SysDesc

⌈An ISignal

- shall not be referenced by DdsCpISignalToDdsTopicMapping in the role iSignal and
- shall not aggregate SOMEIPTransformationISignalProps in the role transformationISignalProps

at the same time.⌋

[constr_9354] Definition of unconditional IPdu Timing

Imposition time: CP: IT_SysDesc

⌈If a TransmissionModeDeclaration has

- no TransmissionModeDeclaration.transmissionModeCondition or all of the aggregated TransmissionModeConditions have the TransmissionModeCondition.dataFilter set to DataFilter.dataFilterType = DataFilterTypeEnum.always, and
- no TransmissionModeDeclaration.modeDrivenTrueCondition defined, and
- no TransmissionModeDeclaration.modeDrivenFalseCondition defined,

then only one TransmissionModeTiming in the role TransmissionModeDeclaration.transmissionModeTrueTiming shall be defined.⌋

[constr_9355] Category of HwElement for EcuPartitionToCoreMapping

Imposition time: CP: IT_EcuExt

⌈The HwElement, which is referenced by EcuPartitionToCoreMapping in the role processingUnit, shall be of category "'ProcessingUnit"'.⌋

[constr_9358] ISignal that references a DataTransformation shall have the dataTypePolicy set to transformingISignal

Imposition time: CP: IT_SysDesc

⌈In a complete model every ISignal that references the DataTransformation in the role dataTransformation shall have the dataTypePolicy set to transformingISignal, ddsSignal, or ddsService.⌋

[constr_9359] Different EcuPartitionToCoreMapping.coreId attribute values shall not be assigned to the same processingUnit

Imposition time: CP: IT_EcuExt

⌈EcuPartitionToCoreMappings that reference the same HwElement in the role processingUnit shall not define different values for the EcuPartitionToCoreMapping.coreId attribute.⌋

[constr_9360] DataMapping not available for elements of PortInterfaces with isService = true

Imposition time: CP: IT_EcuExt

⌈VariableDataPrototypes, ClientServerOperations and Triggers within PortPrototypes typed by PortInterfaces in which the isService attribute is set to true shall not be referenced by a DataMapping.⌋

[constr_9361] FlatInstanceDescriptor.dataPrototype reference restriction

Imposition time: CP: IT_EcuExt

⌈The FlatInstanceDescriptor.dataPrototype reference shall not be used to reference an ArgumentDataPrototype.⌋

[constr_9362] Mutually exclusive existence of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation vs. FlatInstanceDescriptor.ecuExtractReference

Imposition time: CP: IT_EcuExt

⌈If a FlatInstanceDescriptor references an AutosarDataPrototype with role dataPrototype and a BswImplementation in the role bswImplementation then the same FlatInstanceDescriptor shall not reference an AtpFeature in the role ecuExtractReference.⌋

[constr_9363] FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation

Imposition time: CP: IT_EcuExt

⌈If a FlatInstanceDescriptor references an AutosarDataPrototype with role dataPrototype then the FlatInstanceDescriptor shall also reference a BswImplementation in the role FlatInstanceDescriptor.bswImplementation.⌋

[constr_9366] GlobalTimeSlave.followUpTimeoutValue not applicable to GlobalTimeFrSlave

Imposition time: CP: IT_SysDesc

⌈The attribute GlobalTimeSlave.followUpTimeoutValue shall not be defined for a GlobalTimeFrSlave.⌋

[constr_9367] Definition of several local ApplicationEndpoints with identical properties (IP, Port, Protocol) is forbidden

Imposition time: CP: IT_SysDesc

⌈It is not allowed to define several local ApplicationEndpoints (referenced by ProvidedServiceInstances in the role localUnicastAddress or by ConsumedServiceInstances in the role localUnicastAddress) that use the same port number and protocol via tpConfiguration and reference the same NetworkEndpoint in the role networkEndpoint in the context of an EthernetPhysicalChannel.⌋

[constr_9368] Applicable attributes for an Ethernet switch

Imposition time: CP: IT_SysDesc

⌈If a CouplingPort is aggregated by a CouplingElement and that CouplingElement has the attribute couplingType set to the value CouplingElementEnum.switch, then the attributes defined in the column "Switch Port" of [TPS_SYST_02422] are applicable.⌋

[constr_9369] Applicable attributes for an Ethernet node on the classic platform

Imposition time: CP: IT_SysDesc

⌈If a CouplingPort is aggregated by an EthernetCommunicationController and that EthernetCommunicationController is in turn aggregated by an EcuInstance, then the attributes defined in the column "Endpoint" of [TPS_SYST_02422] are applicable.⌋

[constr_9371] Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a RPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the DataPrototype in a RPortPrototype that is mapped to a SystemSignal is compatible to the compuMethods of physicalProps of the SystemSignal if the following conditions apply:

- [constr_1163] (see [2]) that defines the compatibility of CompuMethods is fulfilled,
- [constr_1153] (see [2]) that defines the compatibility of CompuScales of the enclosing CompuMethod is fulfilled,
- [constr_1156] (see [2]) that defines the compatibility of CompuScales which contribute to tabular conversion by having a compuConst,
- [constr_1176] (see [2]) that defines the compatibility of CompuScaless of category LINEAR and RAT_FUNC is fulfilled,
- [constr_1192] (see [2]) that defines the compatibility of CompuScaless of category IDENTICAL to RAT_FUNC or LINEAR is fulfilled,
- [constr_9376] is fulfilled.

⌋

[constr_9372] Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a PPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the DataPrototype in a PPortPrototype that

is mapped to a SystemSignal is compatible to the compuMethods of physicalProps of the SystemSignal if the following conditions apply:

- [constr_1163] (see [2]) that defines the compatibility of CompuMethods is fulfilled,
- [constr_1153] (see [2]) that defines the compatibility of CompuScales of the enclosing CompuMethod is fulfilled,
- [constr_1156] (see [2]) that defines the compatibility of CompuScales which contribute to tabular conversion by having a compuConst,
- [constr_1176] (see [2]) that defines the compatibility of CompuScalesss of category LINEAR and RAT_FUNC is fulfilled,
- [constr_1192] (see [2]) that defines the compatibility of CompuScalesss of category IDENTICAL to RAT_FUNC or LINEAR is fulfilled,
- [constr_9375] is fulfilled.

⌋

[constr_9373] Compatibility of DataConstr between SystemSignal and DataPrototype in a PPortPrototype

Imposition time: CP: IT_SysDesc

⌈The DataConstrs (e.g. the limits) defined in the ApplicationDataType of the DataPrototype in a PPortPrototype shall be within the DataConstrs defined in the SystemSignal.physicalProps of the SystemSignal to which the DataPrototype is mapped.⌋

[constr_9374] Compatibility of DataConstr between SystemSignal and DataPrototype in a RPortPrototype

Imposition time: CP: IT_SysDesc

⌈The DataConstrs (e.g. the limits) defined in the SystemSignal.physicalProps of the SystemSignal to which the DataPrototype in a RPortPrototype is mapped shall be within the DataConstrs defined in the ApplicationDataType of the DataPrototype.⌋

[constr_9375] Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a PPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The set of CompuScales in the CompuMethod defined in the ApplicationDataType of the DataPrototype in a PPortPrototype that is mapped to the SystemSignal shall be a subset of the set of CompuScales defined in the CompuMethod of the SystemSignal.⌋

[constr_9376] Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a RPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The set of CompuScales in the CompuMethod of the SystemSignal shall be a subset of the set of CompuScales defined in the ApplicationDataType of the DataPrototype in a RPortPrototype.⌋

[constr_9377] J1939Node.pduTriggering reference restriction

Imposition time: CP: IT_SysDesc

⌈A J1939Node shall not reference a PduTriggering in the role pduTriggering that in turn references an ISignalIPdu in the role iPdu.⌋

[constr_9378] J1939Nodes on the same EcuInstance shall not share PduTriggerings

Imposition time: CP: IT_SysDesc

⌈A PduTriggering shall not be referenced by several J1939Nodes that are aggregated by the same EcuInstance in the role pduTriggering.⌋

[constr_9379] Consistent mapping of software-component to J1939Node

Imposition time: CP: IT_SysDesc

⌈The value of attribute J1939Node.nodeName.function of a J1939Node referenced by J1939ControllerApplicationToJ1939NodeMapping in the role j1939Node shall be identical to the value of J1939ControllerApplication.functionId.⌋

[constr_9380] Consistent mapping of J1939ControllerApplication to EcuInstance

Imposition time: CP: IT_SysDesc

⌈A SwComponentPrototype that is referenced by a J1939ControllerApplication mapped to a specific J1939Node shall only be mapped to an EcuInstance that in turn owns the same J1939Node.⌋

[constr_9381] J1939Node shall be referenced by a single J1939ControllerApplicationToJ1939NodeMapping

Imposition time: CP: IT_SysDesc

⌈At most one J1939ControllerApplicationToJ1939NodeMapping shall reference a J1939Node in the role J1939ControllerApplicationToJ1939NodeMapping.j1939Node.⌋

[constr_9382] `AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy` shall only target `AppTaskProxy`

Imposition time: CP: `IT_SysDesc`

⌈The reference `AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy` shall only refer to an `OsTaskProxy` that fulfills the definition of an `AppTaskProxy` according to [TPS_SYST_02438]. It shall not target an `EcuTaskProxy` or an `AppAndEcuTaskProxy`.⌋

[constr_9383] `AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy` shall only target `EcuTaskProxy`

Imposition time: CP: `IT_SysDesc`

⌈The reference `AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy` shall only refer to an `OsTaskProxy` that fulfills the definition of an `EcuTaskProxy` according to [TPS_SYST_02438]. It shall not target an `AppTaskProxy` or an `AppAndEcuTaskProxy`.⌋

[constr_9384] Relation between both `CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime` and `CyclicHandlingComDataToOsTaskProxyMapping.offset` and `OsTaskProxy.period`

Imposition time: CP: `IT_SysDesc`

⌈The `CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime` and `CyclicHandlingComDataToOsTaskProxyMapping.offset` (if defined) shall be a positive integer multitude of `period` defined in the `OsTaskProxy` that is referenced by the `CyclicHandlingComDataToOsTaskProxyMapping`.⌋

[constr_9385] Existence of `CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy`

Imposition time: CP: `IT_SysDesc`

⌈For each `CyclicHandlingComDataToOsTaskProxyMapping` the reference `CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy` shall exist.⌋

[constr_9386] Existence of `CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype`

Imposition time: CP: `IT_SysDesc`

⌈For each `CyclicHandlingComDataToOsTaskProxyMapping` the reference `CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype` shall exist at least once.⌋

[constr_9387] One `CyclicHandlingComDataToOsTaskProxyMapping` for every `VariableDataPrototype` with `ReceptionComSpecProps.comHandlerTaskMappingEnabled` = TRUE

Imposition time: CP: `IT_SysDesc`

⌈Every `VariableDataPrototype` that

- is referenced by a ReceiverComSpec with ReceptionComSpecProps.comHandlerTaskMappingEnabled = TRUE and
- is mapped to a SystemSignal or SystemSignalGroup and
- that SystemSignal or SystemSignalGroup has an ISignal or ISignalGroup that is received in the context of the System,

, then there shall be exactly one CyclicHandlingComDataToOsTaskProxyMapping in the context of that System referencing that VariableDataPrototype.⌋

[constr_9388] CanNmNode.txNmPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈Each CanNmNode shall reference at most one NmPdu in the role txNmPdu at the time when the System Description is complete.⌋

[constr_9389] UdpNmNode.txNmPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈Each UdpNmNode shall reference at most one NmPdu in the role txNmPdu at the time when the System Description is complete.⌋

[constr_9390] J1939NmNode.txNmPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈Each J1939NmNode shall reference at most one NmPdu in the role txNmPdu at the time when the System Description is complete.⌋

[constr_9391] FlexrayNmNode.txNmPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈Each FlexrayNmNode shall reference at most four NmPdus in the role txNmPdu at the time when the System Description is complete.⌋

[constr_9395] Each call ISignal shall have exactly one corresponding return ISignal that is associated with the same ClientServerOperation and PhysicalChannel

Imposition time: IT_SysDesc

⌈For each ClientServerToSignalMapping the number of call ISignals

- ISignals referring to a SystemSignal in the role systemSignal and
- that SystemSignal is referenced from the ClientServerToSignalMapping in role callSignal and
- the ISignal is referenced from an ISignalTriggering in role iSignal and
- that ISignalTriggering is aggregated by a PhysicalChannel in role iSignalTriggering

shall be exactly the same as the number of return ISignals

- ISignals referring to a SystemSignal in the role systemSignal and
- that SystemSignal is referenced from the ClientServerToSignalMapping in role returnSignal and
- the ISignal is referenced from an ISignalTriggering in role iSignal and
- that ISignalTriggering is aggregated by the same PhysicalChannel in role iSignalTriggering

on that PhysicalChannel.⌋

[constr_9396] SomeipTpConnection.tpConcurrentProcessingSdu reference restriction

Imposition time: IT_SysDesc

⌈A PduTriggering that is referenced by a SomeipTpConnection in the role tpConcurrentProcessingSdu shall reference an IPdu in the role iPdu.⌋

[constr_9397] Exclusive usage of tpSdu or tpConcurrentProcessingSdu

Imposition time: IT_SysDesc

⌈A SomeipTpConnection shall either define one tpSdu reference or a set of tpConcurrentProcessingSdu references, but not both.⌋

[constr_9398] dataTypePolicy = transformingISignal not applicable for ISignals that are transformed by the ComBasedTransformer

Imposition time: CP: IT_SysDesc

⌈If an ISignal is referenced by an ISignalGroup in the role iSignal and this ISignalGroup references the DataTransformation in the role comBasedSignalGroupTransformation then the ISignal is not allowed to set the dataTypePolicy to transformingISignal.⌋

[constr_9399] Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if no data transformation is used

Imposition time: CP: IT_EcuExt

⌈Either all or a subset of elements of a VariableDataPrototype typed by a composite data type shall be mapped with the SenderReceiverToSignalGroupMapping to SystemSignals of the same SystemSignalGroup if data transformation is not used.⌋

[constr_9400] Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if comBasedSignalGroupTransformation is used

Imposition time: CP: IT_EcuExt

⌈If a ISignalGroup has a reference to the DataTransformation element in the role comBasedSignalGroupTransformation then the ISignalGroup shall refer-

ence a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping that either maps all or a subset of elements of a VariableDataPrototype typed by a composite data type to SystemSignals of this SystemSignalGroup.⌋

[constr_9401] Mapping of VariableDataPrototype typed by a array data type consisting of UINT8 elements to exactly one SystemSignal

Imposition time: CP: IT_EcuExt

⌈The SenderReceiverToSignalGroupMapping is allowed to reference a VariableDataPrototype typed by an array data type if and only if the array data type consists of UINT8 elements.⌋

[constr_9402] PduActivationRoutingGroup with PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast that is aggregated by a ConsumedEventGroup

Imposition time: CP: IT_SysDesc

⌈If a ConsumedEventGroup aggregates at least one PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast, then the aggregating ConsumedServiceInstance shall have a localUnicastAddress reference to a "local" UDP ApplicationEndpoint.⌋

[constr_9403] J1939TpProtocolTypeEnum values fd_bam, fd_bam_cmdt, and fd_cmdt shall only be used on CAN FD clusters

Imposition time: CP: IT_SysDesc

⌈The literals fd_bam, fd_bam_cmdt, and fd_cmdt of a J1939TpConnection.tpProtocolType shall only be used if the J1939TpConnection references a J1939TpNode in the transmitter or receiver role that in turn references a CommunicationConnector that references a CommunicationController with aggregated CanControllerFdConfiguration or CanControllerFdConfigurationRequirements.⌋

[constr_9404] J1939Cluster.usesMultiPg attribute shall only be set to a value on CAN FD clusters

Imposition time: CP: IT_SysDesc

⌈The J1939Cluster.usesMultiPg attribute shall only be set to true if the J1939Cluster also defines a AbstractCanCluster.canFdBaudrate.⌋

[constr_9407] Standardized values for the attribute category of meta-class PortElementToCommunicationResourceMapping

Imposition time: CP: IT_SwCluSysDesc

⌈The following values of the attribute category of meta-class PortElementToCommunicationResourceMapping are reserved by the AUTOSAR standard:

- CLIENT_SERVER_REQUEST: This represents the usage of the PortElementToCommunicationResourceMapping for the purpose to assign an communicationResource to a clientServerOperation used for the request communication path.
- CLIENT_SERVER_RESPONSE: This represents the usage of the PortElementToCommunicationResourceMapping for the purpose to assign a communicationResource to a clientServerOperation used for the response communication path.

⌋

[constr_9408] J1939TpConnection.tpPg in case J1939TpConnection.tpProtocolType is fpp

Imposition time: CP: IT_SysDesc

⌈If J1939TpConnection.tpProtocolType is fpp, only one J1939TpConnection.tpPg shall be configured.⌋

[constr_9421] Exactly two MacSecConnectivityAssociation.couplingPort references for a switched Ethernet connection

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a AbstractMacSecCommunication (MacSecConnectivityAssociation or MacSecBypassedCommunication) refers to a CouplingPort in the role couplingPort that is aggregated by a CouplingElement with couplingType set to switch, then exactly one other couplingPort reference shall reference a CouplingPort that is aggregated by an EthernetCommunicationController.⌋

[constr_9422] Consistent reference to a VLAN using AbstractMacSecCommunication.vlan

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If an EthernetPhysicalChannel is referenced by a AbstractMacSecCommunication (MacSecConnectivityAssociation or MacSecBypassedCommunication) in the role AbstractMacSecCommunication.vlan, then this EthernetPhysicalChannel shall be referenced by the VlanMembership of the CouplingPort that is referenced by the AbstractMacSecCommunication in the role couplingPort.⌋

[constr_9423] Coverage of all VLANs referenced by a CouplingPort by MacSecConnectivityAssociation and MacSecBypassedCommunication

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort is referenced by at least one AbstractMacSecCommunication (MacSecConnectivityAssociation or MacSecBypassedCommunication) in the role couplingPort, then all EthernetPhysicalChannels referenced by the VlanMembership of that CouplingPort shall be referenced by at least one AbstractMacSecCommunication (MacSecConnectivity-

Association or MacSecBypassedCommunication) in the role AbstractMacSecCommunication.vlan.⌋

[constr_9424] 10BASE-T1S MacSecConnectivityAssociation CouplingPortConnection reference consistency

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If at least one CouplingPort that is referenced by a AbstractMacSecCommunication in the role AbstractMacSecCommunication.couplingPort is referenced by a CouplingPortConnection with the nodePort reference, then all CouplingPorts referenced by that AbstractMacSecCommunication in the role AbstractMacSecCommunication.couplingPort shall also be referenced by the same CouplingPortConnection with the nodePort reference.⌋

[constr_9425] Existence of CanSecSecureChannel.csci

Imposition time: CP: IT_SysDesc

⌈For each CanSecConnectivityAssociation the attribute CanSecSecureChannel.csci shall exist.⌋

[constr_9426] Coverage of all CanFrameTriggerings and SoConIPduIdentifiers in the scope of a CanCommunicationConnector by MessageToCanSecSecureChannelMapping and MessageToBypassCanSecCommunicationMapping

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CanFrameTriggering or SoConIPduIdentifier is referenced by a AbstractMessageToSecureChannelMapping (MessageToCanSecSecureChannelMapping or MessageToBypassCanSecCommunicationMapping) in the role canFrameTriggering or soConIPduIdentifier, then every CanFrameTriggering and SoConIPduIdentifier that is aggregated by the same PhysicalChannel where that PhysicalChannel refers to the same CanCommunicationConnector shall be referenced by either a MessageToCanSecSecureChannelMapping or a MessageToBypassCanSecCommunicationMapping in the role canFrameTriggering or soConIPduIdentifier.⌋

[constr_9427] Existence of MessageToCanSecSecureChannelMapping.secureChannel

Imposition time: CP: IT_SysDesc

⌈For each MessageToCanSecSecureChannelMapping the reference MessageToCanSecSecureChannelMapping.secureChannel shall exist.⌋

[constr_9428] Consistent `MessageToCanSecSecureChannelMapping.secureChannel` and `AbstractMessageToSecureChannelMapping.canFrameTriggering` mapping

Imposition time: CP: `IT_SysDesc`

⌈Every `CanFrameTriggering` that is referenced by the `MessageToCanSecSecureChannelMapping` in the role `canFrameTriggering` shall be aggregated by a `PhysicalChannel` and that `PhysicalChannel` shall refer to the same `CanCommunicationConnector` that is referenced by the `MessageToCanSecSecureChannelMapping.secureChannel` in the role `txCommunicationConnector`.⌋

2.7 CP_TPS_TimingExtensions

[constr_4504] Restriction of the `scope` of an `AgeConstraint`

Imposition time: `IT_TimingDescriptionReady`

⌈An `AgeConstraint` may only reference either a:

- `TDEventVariableDataPrototype.tdEventVariableDataPrototypeType==variableDataPrototypeReceived`
- `TDEventTrigger.tdEventTriggerType==triggerActivated`

in the role `scope`⌋

[constr_4505] Specifying minimum and maximum number of occurrences

Imposition time: `IT_TimingDescriptionReady`

⌈The minimum and maximum number of occurrences shall be specified such that the following holds: $0 \leq$ `minNumberOfOccurrences` $\leq$ `maxNumberOfOccurrences`.⌋

[constr_4506] Specifying minimum inter-arrival time and pattern length

Imposition time: `IT_TimingDescriptionReady`

⌈The `minimumInterArrivalTime` and `patternLength` shall be specified such that the following holds: $0 <$ `minimumInterArrivalTime` $\leq$ `patternLength`.⌋

[constr_4507] Specifying pattern length, pattern jitter and patter period

Imposition time: `IT_TimingDescriptionReady`

⌈The `patternLength`, `patternJitter` and `patternPeriod` shall be specified such that the following holds: `patternLength` + `patternJitter` < `patternPeriod`.⌋

[constr_4508] Multiplicity of `TDEventVfbPort.portPrototypeBlueprint`

Imposition time: `IT_TimingViewReady`

⌈The reference `TDEventVfbPort.portPrototypeBlueprint` shall exist only if the immediate parent is `ARPackage.category`==`BLUEPRINT`⌋

[constr_4512] Restriction of `TDEventSwcInternalBehavior.variableAccess`

Imposition time: `IT_TimingViewReady`

⌈The reference `TDEventSwcInternalBehavior.variableAccess` shall exist only if `tdEventSwcInternalBehaviorType==runnableEntityVariableAccess`.⌋

[constr_4513] `SynchronizationTimingConstraint` shall reference at least two events

Imposition time: `IT_TimingDescriptionReady`

⌈In the case, that the `SynchronizationTimingConstraint` is imposed on events then at least two (2) timing description events shall be referenced.⌋

[constr_4514] `SynchronizationTimingConstraint` shall reference at least two event chains

Imposition time: `IT_TimingDescriptionReady`

⌈In the case, that the `SynchronizationTimingConstraint` is imposed on `TimingDescriptionEventChain`s then at least two (2) `TimingDescriptionEventChain`s shall be referenced.⌋

[constr_4515] Orthogonality of `stimulus` and `response` in a `TimingDescriptionEventChain`

Imposition time: `IT_TimingViewReady`

⌈The reference `TimingDescriptionEventChain.stimulus` and the reference `TimingDescriptionEventChain.response` shall not reference the same `TimingDescriptionEvent.shortName`.⌋

[constr_4518] Specifying end-points of a composed `TimingDescriptionEventChain`

Imposition time: `IT_TimingViewReady`

⌈If a `TimingDescriptionEventChain` has > 0 `segment`s: in *that* list of `segment`s:

- at least one `segment.stimulus` shall reference the (parent) `TimingDescriptionEventChain.stimulus` in which it is referenced in the role `segment`
- at least one `segment.response` shall reference the (parent) `TimingDescriptionEventChain.response` in which it is referenced in the role `segment`

⌋

[constr_4519] Specifying `patternLength`

Imposition time: `IT_TimingDescriptionReady`

⌈The `patternLength` shall be specified such that the following holds: $0 \leq max(\texttt{offset}) \leq \texttt{patternLength}$.⌋

[constr_4521] Specifying attribute `synchronizationConstraintType`

Imposition time: IT_TimingDescriptionReady

⌈The attribute synchronizationConstraintType shall be specified if the SynchronizationTimingConstraint is imposed on TimingDescriptionEventChains.⌋

[constr_4522] `SynchronizationTimingConstraint` shall either reference events or event chains

Imposition time: IT_TimingDescriptionReady

⌈The SynchronizationTimingConstraint shall either reference TimingDescriptionEvents or TimingDescriptionEventChains, but not both at the same time.⌋

[constr_4523] Restriction of `maxCycleRepetitions` and `maxSlotsPerCycle` to Repetitive Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, the attributes

- EOCExecutableEntityRefGroup.maxCycleRepetitions
- EOCExecutableEntityRefGroup.maxSlotsPerCycle
- EOCExecutableEntityRefGroup.letInterval
- EOCExecutableEntityRefGroup.triggeringEvent

shall exist only if *that* EOCExecutableEntityRefGroup is aggregated by an ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC in the role orderedElement and *that* EOCExecutableEntityRefGroup is the singularly identifiable ROOT_GROUP (according to [constr_6909])⌋

[constr_4528] The *root* `EOCExecutableEntityRefGroup` shall reference only `EOCExecutableEntityRefGroup`s

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, in the tree of orderedElements, in the singularly identifiable ROOT_GROUP (according to [constr_6909]):

- *that* EOCExecutableEntityRefGroup shall only reference EOCExecutableEntityRefGroups in the role nestedElements
- none of those EOCExecutableEntityRefGroups shall be the ROOT_GROUP (according to [constr_6909])

⌋

[constr_4529] Number of nested elements referenced by the *root* EOCExecutableEntityRefGroup

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, in the tree of orderedElements, in the singularly identifiable ROOT_GROUP (according to [constr_6909]): the number of EOCExecutableEntityRefGroups referenced in the role nestedElements shall be exactly the number given by the attribute maxCycleRepetitions.⌋

[constr_4530] An EOCExecutableEntityRefGroup representing a cycle shall reference only EOCExecutableEntityRefs respectively EOCEventRefs

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, in the tree of orderedElements, a EOCExecutableEntityRefGroup which is NOT the singularly identifiable ROOT_GROUP (according to [constr_6909]): may not reference a EOCExecutableEntityRefGroup in the role nestedElements⌋

[constr_4531] Number of nested elements referenced by EOCExecutableEntityRefGroup representing a cycle

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, in the tree of orderedElements, for a EOCExecutableEntityRefGroup which is NOT the singularly identifiable ROOT_GROUP (according to [constr_6909]): the number of nestedElements referenced by that EOCExecutableEntityRefGroup (representing a cycle) shall be exactly the number given by the attribute maxSlotsPerCycle⌋

[constr_4532] Successor relationship is not self-referencing

Imposition time: IT_TimingViewReady

⌈The target and source of the successor relationships successor and directSuccessor shall not be the same. In other words an EOCExecutableEntityRef and EOCExecutableEntityRefGroup shall not reference itself as its logical or direct successor.⌋

[constr_4533] Maximum number of successor relationships

Imposition time: IT_TimingViewReady

⌈The maximum number of successor relationships, namely successor or directSuccessor:

- between two EOCExecutableEntityRefs
- between two EOCEventRefs
- between two EOCExecutableEntityRefGroups
- between an EOCExecutableEntityRef and an EOCExecutableEntityRefGroup

- between an EOCEventRef and an EOCExecutableEntityRefGroup

is one (1).⌋

[constr_4540] maxCycleRepetitions and maxSlotsPerCycle shall not be zero

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, the attributes:

- maxCycleRepetitions
- maxSlotsPerCycle

shall be greater than zero (0), (if present).⌋

[constr_4541] Multiplicity of EOCExecutableEntityRef.executable in an Ordinary Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In an ExecutionOrderConstraint.executionOrderConstraintType==ordinaryEOC, in the tree of orderedElements, if the orderedElement is a EOCExecutableEntityRef, it shall reference an ExecutableEntity in the role executable.⌋

[constr_4542] Multiplicity of EOCExecutableEntityRef.executable in a Hierarchical Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In an ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC, in the tree of orderedElements, if the orderedElement is a EOCExecutableEntityRef, it shall reference an ExecutableEntity in the role executable.⌋

[constr_4543] Maximum value of minimumInterArrivalTime

Imposition time: IT_TimingDescriptionReady

⌈The minimumInterArrivalTime shall be $\leq$ the period.⌋

[constr_4544] Specifying patternLength, patternJitter and patternPeriod

Imposition time: IT_TimingDescriptionReady

⌈The patternLength, patternJitter and patternPeriod shall be specified such that the following holds: patternLength + patternJitter < patternPeriod.⌋

[constr_4548] Multiplicity of `EOCEventRef.event` in an Ordinary Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In an ExecutionOrderConstraint.executionOrderConstraintType==ordinaryEOC, in the tree of orderedElements, if the orderedElement is a EOCEventRef, it shall reference an AbstractEvent in the role event.⌋

[constr_4549] Multiplicity of `EOCEventRef.event` in a Hierarchical Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In an ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC, in the tree of orderedElements, if the orderedElement is a EOCEventRef, it shall reference an AbstractEvent in the role event.⌋

[constr_4554] Restriction of the referenced `TimingDescriptionEventChain` for a `letInterval`

Imposition time: IT_TimingViewReady

⌈The element EOCExecutableEntityRefGroup.letInterval shall be present only in a ROOT_GROUP (according to [constr_6909])⌋

[constr_4561] Usage of the category value `DISPATCH_ENTRY_POINT` in `TimingDescriptionEvent`

Imposition time: IT_TimingViewReady

⌈The value DISPATCH_ENTRY_POINT of the attribute category of a TimingDescriptionEvent shall be set if and only if the timing description event plays the role of a stimulus event and the corresponding timing description event chain, referencing this timing description event, represents a dispatcher in the context of describing timing of a software cluster.⌋

[constr_4562] Usage of the category value `DISPATCH_EXIT_POINT` in `TimingDescriptionEvent`

Imposition time: IT_TimingViewReady

⌈The value DISPATCH_EXIT_POINT of the attribute category of a TimingDescriptionEvent shall be set if and only if the timing description event plays the role of a response event and the corresponding timing description event chain, referencing this timing description event, represents a dispatcher in the context of describing timing of a software cluster.⌋

[constr_4565] Consistency of `TDCpSoftwareClusterMapping.timingDescription` and `TDCpSoftwareClusterResourceMapping.timingDescription`

Imposition time: IT_TimingViewReady

⌈The references:

- TDCpSoftwareClusterMapping.timingDescription
- TDCpSoftwareClusterResourceMapping.timingDescription

shall (after [constr_6918] has been applied) refer to the same sub-class and category of TimingDescription⌋

[constr_6816] Restricted usage of TimingDescriptionEventChain.isPipeliningPermitted in TimingDescriptionEventChain

Imposition time: IT_TimingViewReady

⌈The attribute isPipeliningPermitted shall only exist if the TimingDescriptionEventChain.category==SL_LET_INTERVAL.⌋

[constr_6817] Restricted usage of TimingDescriptionEvent.clockReference

Imposition time: IT_TimingViewReady

⌈The reference TimingDescriptionEvent.clockReference shall exist if (and only if), the TimingDescriptionEvent is itself referenced:

- in the role stimulus or
- in the role response

by a TimingDescriptionEventChain.category==SL_LET_INTERVAL⌋

[constr_6818] Multiplicity of EventTriggeringConstraint.event

Imposition time: IT_TimingDescriptionReady

⌈For each EventTriggeringConstraint, the reference in the role event shall exist⌋

[constr_6819] Multiplicity of PeriodicEventTriggering.jitter

Imposition time: IT_TimingDescriptionReady

⌈For each PeriodicEventTriggering, the attribute jitter shall exist⌋

[constr_6820] Multiplicity of PeriodicEventTriggering.minimumInterArrivalTime

Imposition time: IT_TimingDescriptionReady

⌈For each PeriodicEventTriggering, the attribute minimumInterArrivalTime shall exist⌋

[constr_6821] Multiplicity of PeriodicEventTriggering.period

Imposition time: IT_TimingDescriptionReady

⌈For each PeriodicEventTriggering, the attribute period shall exist⌋

[constr_6822] Multiplicity of SporadicEventTriggering.maximumInterArrivalTime

Imposition time: IT_TimingDescriptionReady

⌈For each SporadicEventTriggering, the attribute maximumInterArrivalTime shall exist⌋

[constr_6823] Multiplicity of SporadicEventTriggering.minimumInterArrivalTime

Imposition time: IT_TimingDescriptionReady

⌈For each SporadicEventTriggering, the attribute minimumInterArrivalTime shall exist⌋

[constr_6824] Multiplicity of ConcretePatternEventTriggering.patternLength

Imposition time: IT_TimingDescriptionReady

⌈For each ConcretePatternEventTriggering, the attribute patternLength shall exist⌋

[constr_6825] Multiplicity of BurstPatternEventTriggering.maxNumberOfOccurrences

Imposition time: IT_TimingDescriptionReady

⌈For each BurstPatternEventTriggering, the attribute maxNumberOfOccurrences shall exist⌋

[constr_6826] Multiplicity of BurstPatternEventTriggering.minimumInterArrivalTime

Imposition time: IT_TimingDescriptionReady

⌈For each BurstPatternEventTriggering, the attribute minimumInterArrivalTime shall exist⌋

[constr_6827] Multiplicity of BurstPatternEventTriggering.patternLength

Imposition time: IT_TimingDescriptionReady

⌈For each BurstPatternEventTriggering, the attribute patternLength shall exist⌋

[constr_6828] Multiplicity of ArbitraryEventTriggering.minimumDistance

Imposition time: IT_TimingDescriptionReady

⌈For each ArbitraryEventTriggering, the reference in the role minimumDistance shall exist at least once⌋

[constr_6829] Multiplicity of ArbitraryEventTriggering.maximumDistance

Imposition time: IT_TimingDescriptionReady

⌈For each ArbitraryEventTriggering, the reference in the role maximumDistance shall exist at least once⌋

[constr_6830] Multiplicity of ConfidenceInterval.lowerBound

Imposition time: IT_TimingDescriptionReady

⌈For each ConfidenceInterval, the attribute lowerBound shall exist⌋

[constr_6831] Multiplicity of ConfidenceInterval.propability

Imposition time: IT_TimingDescriptionReady

⌈For each ConfidenceInterval, the attribute propability shall exist⌋

[constr_6832] Multiplicity of ConfidenceInterval.upperBound

Imposition time: IT_TimingDescriptionReady

⌈For each ConfidenceInterval, the attribute upperBound shall exist⌋

[constr_6833] Multiplicity of ExecutionOrderConstraint.orderedElement

Imposition time: IT_TimingDescriptionReady

⌈For each ExecutionOrderConstraint, the attribute orderedElement shall exist at least once⌋

[constr_6834] Multiplicity of EOCExecutableEntityRefGroup.nestedElement

Imposition time: IT_TimingDescriptionReady

⌈For each EOCExecutableEntityRefGroup, the reference in the role nestedElement shall exist at least once⌋

[constr_6835] Multiplicity of ExecutionTimeConstraint.executionTimeType

Imposition time: IT_TimingViewReady

⌈For each ExecutionTimeConstraint, the attribute executionTimeType shall exist⌋

[constr_6836] Multiplicity of ExecutionTimeConstraint.executable

Imposition time: IT_TimingViewReady

⌈For each ExecutionTimeConstraint, the reference to ExecutableEntity in the role executable shall exist⌋

[constr_6837] Multiplicity of LatencyTimingConstraint.latencyConstraintType

Imposition time: IT_TimingDescriptionReady

⌈For each LatencyTimingConstraint, the attribute LatencyTimingConstraint.latencyConstraintType shall exist⌋

[constr_6838] Multiplicity of `LatencyTimingConstraint.maximum`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `LatencyTimingConstraint`, the attribute `LatencyTimingConstraint.maximum` shall exist⌋

[constr_6839] Multiplicity of `LatencyTimingConstraint.minimum`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `LatencyTimingConstraint`, the attribute `LatencyTimingConstraint.minimum` shall exist⌋

[constr_6841] Multiplicity of `LatencyTimingConstraint.scope`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `LatencyTimingConstraint`, the reference `LatencyTimingConstraint.scope` shall exist⌋

[constr_6842] Multiplicity of `OffsetTimingConstraint.maximum`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `OffsetTimingConstraint`, the attribute `maximum` shall exist⌋

[constr_6843] Multiplicity of `OffsetTimingConstraint.minimum`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `OffsetTimingConstraint`, the attribute `minimum` shall exist⌋

[constr_6844] Multiplicity of `OffsetTimingConstraint.source`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `OffsetTimingConstraint`, the reference in the role `source` shall exist⌋

[constr_6845] Multiplicity of `OffsetTimingConstraint.target`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `OffsetTimingConstraint`, the reference in the role `target` shall exist⌋

[constr_6846] Multiplicity of `SynchronizationTimingConstraint.synchronizationConstraintType`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `SynchronizationTimingConstraint`, the attribute `synchronizationConstraintType` shall exist⌋

[constr_6847] Multiplicity of `SynchronizationTimingConstraint.tolerance`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `SynchronizationTimingConstraint`, the attribute `tolerance` shall exist⌋

[constr_6849] Multiplicity of SystemTiming.system

Imposition time: IT_TimingViewReady

⌈For each SystemTiming, the reference to System in the role system shall exist⌋

[constr_6850] Multiplicity of BswModuleTiming.behavior

Imposition time: IT_TimingViewReady

⌈For each BswModuleTiming, the reference to BswInternalBehavior in the role behavior shall exist⌋

[constr_6851] Multiplicity of BswCompositionTiming.implementation

Imposition time: IT_TimingViewReady

⌈For each BswCompositionTiming, the reference to BswImplementation in the role implementation shall exist at least once⌋

[constr_6852] Multiplicity of EcuTiming.ecuConfiguration

Imposition time: IT_TimingViewReady

⌈For each EcuTiming, the reference to EcucValueCollection in the role ecuConfiguration shall exist at least once⌋

[constr_6853] Multiplicity of ModeInBswInstanceRef.contextModeDeclarationGroupPrototype

Imposition time: IT_TimingViewReady

⌈For each ModeInBswInstanceRef, the reference to ModeDeclarationGroupPrototype in the role contextModeDeclarationGroupPrototype shall exist at least once.⌋

[constr_6854] Multiplicity of ModeInBswInstanceRef.targetModeDeclaration

Imposition time: IT_TimingViewReady

⌈For each ModeInBswInstanceRef, the reference to ModeDeclaration in the role targetModeDeclaration shall exist at least once.⌋

[constr_6855] Multiplicity of ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype

Imposition time: IT_TimingViewReady

⌈For each ModeInSwcInstanceRef, the reference to ModeDeclarationGroupPrototype in the role contextModeDeclarationGroupPrototype shall exist at least once.⌋

[constr_6856] Multiplicity of ModeInSwcInstanceRef.contextPort

Imposition time: IT_TimingViewReady

⌈For each ModeInSwcInstanceRef, the reference to PortPrototype in the role contextPort shall exist at least once.⌋

[constr_6857] Multiplicity of ModeInSwcInstanceRef.targetModeDeclaration

Imposition time: IT_TimingViewReady

⌈For each ModeInSwcInstanceRef, the reference to ModeDeclaration in the role targetModeDeclaration shall exist at least once.⌋

[constr_6858] Multiplicity of TDEventBswInternalBehavior.tdEventBswInternalBehaviorType

Imposition time: IT_TimingViewReady

⌈The attribute TDEventBswInternalBehavior.tdEventBswInternalBehaviorType shall exist⌋

[constr_6859] Multiplicity of TDEventBswInternalBehavior.bswModuleEntity

Imposition time: IT_TimingViewReady

⌈The attribute TDEventBswInternalBehavior.bswModuleEntity shall exist⌋

[constr_6860] Multiplicity of TDEventBswModule.tdEventBswModuleType

Imposition time: IT_TimingViewReady

⌈The attribute TDEventBswModule.tdEventBswModuleType shall exist⌋

[constr_6861] Multiplicity of TDEventBswModule.bswModuleEntry

Imposition time: IT_TimingViewReady

⌈The reference TDEventBswModule.bswModuleEntry shall exist⌋

[constr_6862] Multiplicity of TDEventBswModeDeclaration.tdEventBswModeDeclarationType

Imposition time: IT_TimingViewReady

⌈The reference tdEventBswModeDeclarationType shall exist⌋

[constr_6863] Multiplicity of TDEventBswModeDeclaration.modeDeclaration

Imposition time: IT_TimingViewReady

⌈The reference TDEventBswModeDeclaration.modeDeclaration shall exist⌋

[constr_6864] Multiplicity of TDEventISignal.tdEventType

Imposition time: IT_TimingViewReady

⌈For each TDEventISignal, the attribute tdEventType shall exist⌋

[constr_6865] Multiplicity of TDEventISignal.iSignal

Imposition time: IT_TimingViewReady

⌈For each TDEventISignal, the reference to ISignal in the role iSignal shall exist⌋

[constr_6866] Multiplicity of TDEventISignal.physicalChannel

Imposition time: IT_TimingViewReady

⌈For each TDEventISignal, the reference to PhysicalChannel in the role physicalChannel shall exist⌋

[constr_6867] Multiplicity of TDEventIPdu.tdEventType

Imposition time: IT_TimingViewReady

⌈For each TDEventIPdu, the attribute tdEventType shall exist⌋

[constr_6868] Multiplicity of TDEventIPdu.iPdu

Imposition time: IT_TimingViewReady

⌈For each TDEventIPdu, the reference to IPdu in the role iPdu shall exist⌋

[constr_6869] Multiplicity of TDEventIPdu.physicalChannel

Imposition time: IT_TimingViewReady

⌈For each TDEventIPdu, the reference to PhysicalChannel in the role physical-Channel shall exist⌋

[constr_6870] Multiplicity of TDEventFrame.tdEventType

Imposition time: IT_TimingViewReady

⌈For each TDEventFrame, the attribute tdEventType shall exist⌋

[constr_6871] Multiplicity of TDEventFrame.frame

Imposition time: IT_TimingViewReady

⌈For each TDEventFrame, the reference to Frame in the role frame shall exist⌋

[constr_6872] Multiplicity of TDEventFrame.physicalChannel

Imposition time: IT_TimingViewReady

⌈For each TDEventFrame, the reference to PhysicalChannel in the role physicalChannel shall exist⌋

[constr_6873] Multiplicity of TDEventFrameEthernet.tdEventType

Imposition time: IT_TimingViewReady

⌈For each TDEventFrameEthernet, the attribute tdEventType shall exist⌋

[constr_6874] Multiplicity of TDHeaderIdRange.maxHeaderId

Imposition time: IT_TimingViewReady

⌈For each TDHeaderIdRange, the attribute maxHeaderId shall exist⌋

[constr_6875] Multiplicity of TDHeaderIdRange.minHeaderId

Imposition time: IT_TimingViewReady

⌈For each TDHeaderIdRange, the attribute minHeaderId shall exist⌋

[constr_6876] Multiplicity of TDEventCycleStart.cycleRepetition

Imposition time: IT_TimingViewReady

⌈For each TDEventCycleStart, the attribute cycleRepetition shall exist⌋

[constr_6877] Multiplicity of TDEventFrClusterCycleStart.frCluster

Imposition time: IT_TimingViewReady

⌈For each TDEventFrClusterCycleStart, the attribute frCluster shall exist⌋

[constr_6879] Multiplicity of TDEventOccurrenceExpression.formula

Imposition time: IT_TimingDescriptionReady

⌈The aggregation TDEventOccurrenceExpression.formula shall exist⌋

[constr_6880] Multiplicity of AutosarVariableInstance.variableInstance

Imposition time: IT_TimingDescriptionReady

⌈The instance reference AutosarVariableInstance.variableInstance shall exist⌋

[constr_6881] Multiplicity of AutosarOperationArgumentInstance.operationArgumentInstance

Imposition time: IT_TimingDescriptionReady

⌈The instance reference AutosarOperationArgumentInstance.operationArgumentInstance shall exist⌋

[constr_6882] Multiplicity of TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType

Imposition time: IT_TimingViewReady

⌈The attribute TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType shall exist⌋

[constr_6883] Multiplicity of TDEventSwcInternalBehavior.runnable

Imposition time: IT_TimingViewReady

⌈The reference TDEventSwcInternalBehavior.runnable shall exist⌋

[constr_6884] Multiplicity of TDEventSwcInternalBehaviorReference.referencedTDEventSwc

Imposition time: IT_TimingViewReady

⌈The reference TDEventSwcInternalBehaviorReference.referencedTDEventSwc shall exist⌋

[constr_6885] Multiplicity of TDEventVfbPort.isExternal

Imposition time: IT_TimingViewReady

⌈For each TDEventVfbPort, the attribute isExternal shall exist⌋

[constr_6886] Multiplicity of TDEventVfbReference.referencedTDEventVfb

Imposition time: IT_TimingViewReady

⌈For each TDEventVfbReference, the reference to TDEventVfb in the role referencedTDEventVfb shall exist⌋

[constr_6887] Multiplicity of TDEventVariableDataPrototype.tdEventVariableDataPrototypeType

Imposition time: IT_TimingViewReady

⌈For each TDEventVariableDataPrototype, the attribute tdEventVariableDataPrototypeType shall exist⌋

[constr_6888] Multiplicity of TDEventVariableDataPrototype.dataElement

Imposition time: IT_TimingViewReady

⌈For each TDEventVariableDataPrototype, the reference to VariableDataPrototype in the role dataElement shall exist⌋

[constr_6889] Multiplicity of TDEventOperation.tdEventOperationType

Imposition time: IT_TimingViewReady

⌈For each TDEventOperation, the attribute tdEventOperationType shall exist⌋

[constr_6890] Multiplicity of TDEventOperation.operation

Imposition time: IT_TimingViewReady

⌈For each TDEventOperation, the reference to ClientServerOperation in the role operation shall exist⌋

[constr_6891] Multiplicity of TDEventModeDeclaration.tdEventModeDeclarationType

Imposition time: IT_TimingViewReady

⌈For each TDEventModeDeclaration, the attribute tdEventModeDeclarationType shall exist⌋

[constr_6892] Multiplicity of TDEventModeDeclaration.modeDeclaration

Imposition time: IT_TimingViewReady

⌈For each TDEventModeDeclaration, the reference to ModeDeclarationGroupPrototype in the role modeDeclaration shall exist⌋

[constr_6893] Multiplicity of TDEventTrigger.tdEventTriggerType

Imposition time: IT_TimingViewReady

⌈For each TDEventTrigger, the attribute tdEventTriggerType shall exist⌋

[constr_6894] Multiplicity of `TDEventTrigger.trigger`

Imposition time: `IT_TimingViewReady`

⌈For each `TDEventTrigger`, the reference to `Trigger` in the role `trigger` shall exist⌋

[constr_6895] Multiplicity of `TimingDescriptionEventChain.response`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `TimingDescriptionEventChain`, the reference in the role `response` shall exist⌋

[constr_6896] Multiplicity of `TimingDescriptionEventChain.stimulus`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `TimingDescriptionEventChain`, the reference in the role `stimulus` shall exist⌋

[constr_6898] Multiplicity of `ConcretePatternEventTriggering.offset`

Imposition time: `IT_TimingDescriptionReady`

⌈For each `ConcretePatternEventTriggering`, the attribute `offset` shall exist⌋

[constr_6899] Multiplicity of `ModeInSwcInstanceRef.base`

Imposition time: `IT_TimingViewReady`

⌈For each `ModeInSwcInstanceRef`, the reference to `SwComponentType` in the role `base` shall exist at least once.⌋

[constr_6900] Dual existence of `TDEventVfb.portPrototype` and `TDEventVfb.portPrototypeBlueprint`

Imposition time: `IT_TimingViewReady`

⌈The reference `TDEventVfbPort.portPrototype` and `TDEventVfbPort.portPrototypeBlueprint` shall not co-exist in a model⌋

[constr_6901] Multiplicity of `TDEventBsw.bswModuleDescription`

Imposition time: `IT_TimingViewReady`

⌈The attribute `TDEventBsw.bswModuleDescription` shall exist⌋

[constr_6906] Conformity of `stimulus` and `response` in a `TimingDescriptionEventChain`

Imposition time: `IT_TimingDescriptionReady`

⌈The `TimingDescriptionEvent`s referenced in the roles `stimulus` and `response` shall be of the same sub-class (of `TimingDescriptionEvent`)⌋

[constr_6908] Restriction of EOCExecutableEntityRefGroup.letDataExchangeParadigm

Imposition time: IT_TimingViewReady

⌈The attribute letDataExchangeParadigm shall exist only if the letInterval in the same same EOCExecutableEntityRefGroup references a TimingDescriptionEventChain.category==LET_INTERVAL⌋

[constr_6909] Singleton ROOT_GROUP in an Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In either:

- ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC
- ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC

in the tree of orderedElements, there shall be only **one** EOCExecutableEntityRefGroup.category==ROOT_GROUP⌋

[constr_6910] Referencing FROM a ROOT_GROUP in an Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In either:

- ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC
- ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC

in the tree of orderedElements, in the singularly identifiable ROOT_GROUP (according to [constr_6909]):

- the successor shall not exist
- the directSuccessor shall not exist

⌋

[constr_6911] Referencing TO a ROOT_GROUP in an Execution Order Constraint

Imposition time: IT_TimingViewReady

⌈In either:

- ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC
- ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC

in the tree of orderedElements, the singularly identifiable ROOT_GROUP (according to [constr_6909]) shall not be referenced in the role:

- successor by another EOCExecutableEntityRefGroup
- directSuccessor by another EOCExecutableEntityRefGroup
- nestedElement by another EOCExecutableEntityRefGroup

⌋

[constr_6912] Mandatory specification of LET interval recurrence

Imposition time: IT_TimingViewReady

⌈For a TDEventComplex.category==LET_RELEASE, there shall exist a PeriodicEventTriggering referencing it (*that* same TDEventComplex) in the role event⌋

[constr_6913] Restriction on RTEEvents used in an LET interval

Imposition time: IT_TimingViewReady

⌈An EOCExecutableEntityRefGroup which references a TimingDescriptionEventChain.category==LET_INTERVAL in the role letInterval **and** transitively references an EOCEventRef in the role successor **or** references an EOCEventRef in the role nestedElement, those EOCEventRefs shall reference either:

- TimingEvent in the role event or
- BswTimingEvent in the role event

⌋

[constr_6914] Restriction of the portPrototype context of an AgeConstraint

Imposition time: IT_TimingDescriptionReady

⌈An AgeConstraint.scope.portPrototype shall reference only sub-classes of AbstractRequiredPortPrototype⌋

[constr_6915] Affinity of ISignal in TDEventISignal

Imposition time: IT_TimingViewReady

⌈The referenced ISignal in the role TDEventISignal.iSignal shall exist also in the list of iSignals aggregated by TDEventISignal.physicalChannel.iSignalTriggering⌋

[constr_6916] Affinity of Frame in TDEventFrame

Imposition time: IT_TimingViewReady

⌈The referenced Frame in the role TDEventFrame.frame shall exist also in the list of frames aggregated by TDEventFrame.physicalChannel.frameTriggering⌋

[constr_6917] Affinity of `IPdu` in `TDEventIPdu`

Imposition time: IT_TimingViewReady

⌈The referenced IPdu in the role TDEventIPdu.iPdu shall exist also in the list of iPdus aggregated by TDEventIPdu.physicalChannel.pduTriggering⌋

[constr_6918] Referenced `TimingDescription`s in `TDCpSoftwareClusterMapping` and `TDCpSoftwareClusterResourceMapping`

Imposition time: IT_TimingViewReady

⌈The references:

- TDCpSoftwareClusterMapping.timingDescription
- TDCpSoftwareClusterResourceMapping.timingDescription

shall refer to either:

- TDEventComplex.category==DISPATCH_ENTRY_POINT, or
- TimingDescriptionEventChain.category==LET_INTERVAL

⌋

[constr_6919] Referenced `CpSoftwareCluster` of `TDCpSoftwareClusterMapping`

Imposition time: IT_TimingViewReady

⌈Each TDCpSoftwareClusterMapping.provider reference shall refer to a CpSoftwareCluster.category==HOST_SOFTWARE_CLUSTER and each TDCpSoftwareClusterMapping.requestor reference shall refer to a CpSoftwareCluster.category==APPLICATION_SOFTWARE_CLUSTER⌋

[constr_6920] Multiplicity of `LatencyTimingConstraint.minimum` used in an LET interval

Imposition time: IT_TimingViewReady

⌈For a LatencyTimingConstraint with:

- latencyConstraintType==reaction
- scope.category==LET_INTERVAL

the attribute minimum shall not exist.⌋

[constr_6921] Disallow `TimingDescriptionEventChain` segmental circular-referencing

Imposition time: IT_TimingViewReady

⌈A TimingDescriptionEventChain.segment shall never reference the (parent) TimingDescriptionEventChain in which it is referenced in the role segment.⌋

[constr_6922] Restriction of `System.category` in a `SystemTiming` view

Imposition time: IT_TimingViewReady

⌈The reference SystemTiming.system shall refer only to one of the following System.categorys from [TPS_SYST_01003]:

- SYSTEM_DESCRIPTION
- SYSTEM_EXTRACT
- ECU_SYSTEM_DESCRIPTION
- ECU_EXTRACT
- SW_CLUSTER_SYSTEM_DESCRIPTION

⌋

[constr_6923] Restriction of `System.category` in a `VfbTiming` view

Imposition time: IT_TimingViewReady

⌈The SwComponentType referenced by VfbTiming.component shall be restricted to a SwComponentType which is referenced by a SwComponentPrototype.type and that SwComponentPrototype is aggregated by a CompositionSwComponentType.component and that CompositionSwComponentType is transitively referenced by a RootSwCompositionPrototype.softwareComposition and that RootSwCompositionPrototype is aggregated by a System.rootSoftwareComposition where the System.category shall be one of those in [constr_6922].⌋

[constr_6924] Multiplicity of `TDEventOccurrenceExpression.function`

Imposition time: IT_TimingDescriptionReady

⌈The attribute TDEventOccurrenceExpression.function shall exist⌋

[constr_6926] Restriction of `successor` and `directSuccessor`

Imposition time: IT_TimingViewReady

⌈In a executionOrderConstraintType==repetitiveEOC, in the tree of orderedElements:

- EOCExecutableEntityRefGroup.directSuccessor
- EOCExecutableEntityRefGroup.successor
- EOCExecutableEntityRef.directSuccessor
- EOCExecutableEntityRef.successor
- EOCEventRef.directSuccessor
- EOCEventRef.successor

shall not exist⌋

[constr_6927] Exclusivity of either `EOCExecutableEntityRef` or `EOCEventRef` in an `ExecutionOrderConstraint`

Imposition time: IT_TimingViewReady

⌈In a ExecutionOrderConstraint, in the tree of orderedElements, dual use of EOCEventRef and EOCExecutableEntityRef is not permitted, such that only either:

- EOCEventRef
  - via aggregation ExecutionOrderConstraint.orderedElements
- EOCExecutableEntityRef
  - via aggregation ExecutionOrderConstraint.orderedElements
  - via reference ExecutionOrderConstraint.EOCExecutableEntityRefGroup.nestedElements

is permitted⌋

[constr_6928] Exclusivity of `component` and `bswModuleInstance` in an `EOCExecutableEntityRef`

Imposition time: IT_TimingDescriptionReady

⌈The instance reference EOCExecutableEntityRef.component and the reference EOCExecutableEntityRef.bswModuleInstance shall not co-exist⌋

[constr_6929] Exclusivity of `component` and `bswModuleInstance` in an `EOCEventRef`

Imposition time: IT_TimingDescriptionReady

⌈The instance reference EOCEventRef.component and the reference EOCEventRef.bswModuleInstance shall not co-exist⌋

[constr_6930] Restriction of `ExecutionTimeConstraint.component`

Imposition time: IT_TimingViewReady

⌈The attribute ExecutionTimeConstraint.component shall exist only if ExecutionTimeConstraint.executable references a RunnableEntity.⌋

[constr_6931] Affinity of `BswModuleEntity` in GET/RT

Imposition time: IT_TimingDescriptionReady

⌈If, after [TPS_TIMEX_00141] is applied, the LatencyTimingConstraint.scope.stimulus and LatencyTimingConstraint.scope.response reference a TDEventBswInternalBehavior then the same BswModuleEntity shall be referenced by both.⌋

[constr_6932] Affinity of `RunnableEntity` in GET/RT

Imposition time: `IT_TimingDescriptionReady`

⌈If, after [TPS_TIMEX_00141] is applied, the `LatencyTimingConstraint.scope.stimulus` and `LatencyTimingConstraint.scope.response` reference a `TDEventSwcInternalBehavior` then the same `RunnableEntity` shall be referenced by both.⌋

2.8 FO_TPS_GenericStructureTemplate

[constr_2501] Blueprint of blueprints are not supported ⌈Note that objects modeled particularly as a "blueprint" (e.g. `PortPrototypeBlueprint`) also live in a package of category `BLUEPRINT`. Strictly speaking this means that they can be "blueprints" of "blueprints". This indirection is not intended and not supported.⌋

[constr_2502] Merged model shall be compliant to the meta-model ⌈A model merged from ≪`atpSplitable`≫ elements shall adhere to the constraints of the *meta-model*. This concerns, for example, the lower and upper multiplicities of model elements. Note that the applied constraints depend on the imposition times.⌋

[constr_2504] Constraint to `bindingTime`

Imposition time: CP: `IT_Variant`

⌈The tag `vh.latestBindingTime` *constraints* the value of the attribute `bindingTime` from [TPS_GST_00190]. Hence, it defines the latest point in methodology which is allowed as value for `bindingTime` of this particular application of ≪`atpVariation`≫.⌋

[constr_2505] Multiplicity after binding

Imposition time: CP: `IT_Variant`

⌈*if* Phase ≥ `{partRole}.BindingTime` *then* number of `{partRole}`'s = n⌋

[constr_2507] `EvaluatedVariantSet` shall not refer to itself

Imposition time: CP: `IT_Variant`

⌈An `EvaluatedVariantSet` shall not refer to itself directly or via other `EvaluatedVariantSet`.⌋

[constr_2508] The `shortName` shall be unique in its name space ⌈The content of `shortName` needs to be unique (case insensitive) within a the parent given name space.

Note that the check for uniqueness of `shortName` shall be performed case insensitively. This supports the good practice that names should not differ in upper / lower case only which would cause a lot of confusion.

The term "case insensitive" indicates that the characters in the sets

{a b c d e f g h i j k l m n o p q r s t u v w x y z}
{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}

are respectively considered to be the same. In other words case-insensitive check for uniqueness of shortNames results in the fact that e.g. elements with shortName "X" and "x" are considered the same and shall not exist in the same name space.⌋

[constr_2509] Uniqueness of ReferenceBase.shortLabel in the scope of an ARPackage ⌈The shortLabel of any given ReferenceBase defined in the scope of an ARPackage shall be unique within the scope of the enclosing ARPackage.⌋

[constr_2511] Named reference bases shall be available ⌈If there is a relative references, then one of the containing packages shall have a referenceBase with a shortLabel equal to the base of the reference.⌋

[constr_2512] shortName uniqueness constraint for variants

Imposition time: CP: IT_Variant

⌈shortName + shortLabel of a variant element shall be unique within the name space established by the surrounding Identifiable.⌋

[constr_2514] shortLabel in VariationPoint shall be unique

Imposition time: CP: IT_Variant

⌈The combination of shortName and shortLabel shall be unique within the next enclosing Identifiable {WholeClass}.⌋

[constr_2515] Categories of packages shall not conflict ⌈

	child + category (also indirect children)				
parent category	empty	BLUEPRINT	STANDARD	custom1	custom2
empty	ok	ok	ok	ok	ok
BLUEPRINT	ok	ok	conflict	conflict	conflict
STANDARD	ok	conflict	ok	conflict	conflict
custom1	ok	conflict	conflict	ok	conflict
custom2	ok	conflict	conflict	conflict	ok
	target package category (if category is empty, then the parent category applies)				
category of package that contains reference source element (if category is empty, then the parent category applies)	empty	BLUEPRINT	STANDARD	custom1	custom2
empty	ok	ok	ok	ok	ok
BLUEPRINT	ok	ok	ok	conflict	conflict
STANDARD	ok	conflict	ok	conflict	conflict
custom1	ok	ok	ok	ok	ok

▽

△

custom2	ok	ok	ok	ok	ok

Upper part of the table: If a non empty category is defined for a package, then all sub packages shall have empty category or the same category.
Lower part of the table: Additionally, the "Rules for references between elements in packages with specific categories" shall apply.

⌋

[constr_2516] Return type of an `AttributeValueVariationPoint`

Imposition time: CP: `IT_Variant`

⌈When such a formula is evaluated by a software tool, then the return value of the formula is shall be one of:

- compatible to the type of the attribute
- numerical value which can be taken up to look in the `EnumerationMappingTable` [11] if the attribute in the pure meta-model represents an enumeration.

⌋

[constr_2517] `postBuildVariantCondition` only for PostBuild

Imposition time: CP: `IT_Variant`

⌈Aggregation of `PostBuildVariantCondition` in `VariationPoint` is only allowed if the annotated model states `vh.latestBindingTime` to `PostBuild`.⌋

[constr_2518] Binding time is constrained

Imposition time: CP: `IT_Variant`

⌈Note that this binding time is again constrained by the value of the tag `vh.latestBindingTime`.⌋

[constr_2519] `PredefinedVariant`s need to be consistent

Imposition time: CP: `IT_Variant`

⌈If a `PredefinedVariant` plus its `includedVariant`s references more than one `SwSystemconstantValueSet` all `value` attributes in `SwSystemconstValue`s for a particular `SwSystemconst` shall be identical.⌋

[constr_2521] The `shortLabel` in `AttributeValueVariationPoint` shall be unique

Imposition time: CP: `IT_Variant`

⌈The `shortLabel` shall be unique (case insensitive) within the next enclosing `Identifiable` and is used to individually address variation points in the *variant-rich M1 model*.

Note that the check for uniqueness of `shortLabel` shall be performed case insensitively. This supports the good practice that `shortLabel`s should not differ in upper / lower case only which would cause a lot of confusion.

The term 'case insensitive' indicates that the characters in the sets

```
{a b c d e f g h i j k l m n o p q r s t u v w x y z}
{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}
```

are respectively considered to be the same. In other words case-insensitive check for uniqueness of `shortLabel` results in the fact that e.g. elements with `shortLabel` 'X' and 'x' are considered the same and shall not exist in the same context.⌋

[constr_2522] Notes should not be nested ⌈Note even if it is possible to nest notes it is not recommended to do so, since it might lead to problems with the rendering of the note icon.⌋

[constr_2523] Used languages need to be consistent ⌈The used languages of an AUTOSAR file are specified in the top level `adminData`. All other elements shall be provided in the languages specified for the document.⌋

[constr_2533] Documentation context is either a feature or an identifiable ⌈One particular `DocumentationContext` shall be either a feature or an identifiable but not both at the same time. If this is desired, one should create multiple `DocumentationContext`.⌋

[constr_2534] Practically `UnlimitedInteger` shall be limited such that it fits into 64 bit. ⌈If a signed value is represented the min value can be down to -9223372036854775808 (0x800000000000000014) and the max value can be up to 9223372036854775807 (`0x7fffffffffffffff`).⌋

[constr_2538] Global reference is limited to certain elements ⌈The ability to perform a global reference is limited to

- `Chapter`,
- `Topic1`,
- `Caption`,
- `Traceable`,
- `XrefTarget`,
- `Std`,
- `Xdoc`,
- `Xfile`

⌋

[constr_2542] Value of `AttributeValueVariationPoint.bindingTime`

Imposition time: CP: `IT_Variant`

⌈The value of `AttributeValueVariationPoint.bindingTime` shall be less than or equal to the value of the tag `vh.latestBindingTime` attached to the attribute according to the order defined by [TPS_GST_00450].⌋

[constr_2547] Ordered collections cannot be split into different partial models ⌈Ordered collections cannot be split. In other words: Contrary to the semantics of unordered collections - which can be distributed between partial models - ordered collections can only be placed as a whole in one of the partial models. Otherwise the undefined merge order would violate the semantics of the collections.⌋

[constr_2557] No `VariationPoint`s where `vh.latestBindingTime` set to `BlueprintDerivationTime` in system configurations

Imposition time: CP: `IT_Variant`

⌈Blueprints are **not** part of a system configuration. In consequence of this, in a system configuration there shall be no `VariationPoint` where `vh.latestBindingTime` is restricted to `BlueprintDerivationTime` by the meta-model.⌋

[constr_2558] If `vh.latestBindingTime` is `BlueprintDerivationTime` then there shall only be `blueprintCondition` or `formalBlueprintGenerator` respectively `blueprintValue`

Imposition time: CP: `IT_Variant`

⌈`VariationPoint`s with `vh.latestBindingTime` restricted to `BlueprintDerivation` shall not have `swSyscond` nor `postBuildVariantCondition`.⌋

[constr_2559] No nested `VariationPoint`

Imposition time: CP: `IT_Variant`

⌈As `blueprintCondition` is a `DocumentationBlock` it could again contain `VariationPoint`s and therefore would allow nesting of `VariationPoint`s. This is not intended and shall not be used.⌋

[constr_2567] Undefined Value in Attribute Value Blueprints

Imposition time: CP: `IT_BlueprintCreation`

⌈If a `blueprintValue` is specified, then the `value` defined by the `AttributeValueVariationPoint` is not used and should therefore at least contain one term `undefined` which is to be refined when deriving objects from this blueprint.⌋

[constr_2572] Unique Control of Document Languages ⌈The settings for multiple languages are specified in the top-Level `AdminData` only⌋

[constr_2575] `blueprintValue` in blueprints only

Imposition time: CP: `IT_BlueprintCreation`

⌈`blueprintValue` is only allowed in blueprints and may not be present in a system description.⌋

[constr_2577] Binding Time in Aggregation Pattern

Imposition time: CP: `IT_Variant`

⌈Within `VariationPoint`, the class `ConditionByFormula` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the aggregation see [TPS_GST_00190], [TPS_GST_00220], [TPS_GST_00221]):

`ConditionByFormula.bindingTime` $\leq$ *aggregation*.`vh.latestBindingTime`⌋

[constr_2578] Binding Time in Association Pattern

Imposition time: CP: `IT_Variant`

⌈Within `VariationPoint`, the class `ConditionByFormula` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the association (see [TPS_GST_00190], [TPS_GST_00220],[TPS_GST_00221]):

`ConditionByFormula.bindingTime` $\leq$ *association*.`vh.latestBindingTime`⌋

[constr_2580] Binding Time in Property Set Pattern

Imposition time: CP: `IT_Variant`

⌈The meta-class `VariationPoint` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the meta-class which is marked as ≪`atpVariation`≫ (see [TPS_GST_00190], [TPS_GST_00220], [TPS_GST_00221]):

`VariationPoint.bindingTime` $\leq$ *meta class*.`vh.latestBindingTime`⌋

[constr_2581] Default life cycle state shall be defined properly ⌈`defaultLcState` in `LifeCycleInfoSet` shall reference to a `lcState` defined in the `LifeCycleStateDefinitionGroup` referenced by `usedLifeCycleStateDefinitionGroup`.⌋

[constr_2583] Used life cycle state shall be defined properly ⌈`defaultLcState` in `LifeCycleInfo` shall reference to a `lcState` defined in the `LifeCycleStateDefinitionGroup` referenced by `usedLifeCycleStateDefinitionGroup` of the containing `LifeCycleInfoSet`.⌋

[constr_2585] `LifeCycleInfo` shall be unambiguous ⌈Within one particular `LifeCycleInfoSet` `lifeCycleInfo`.`lcObject` shall be unique. This ensures that the association of a `LifeCycleState` to a `Referrable` is unambiguous.

This constraint applies for a particular point in time under consideration of the period of viability according to [TPS_GST_00244].⌋

[constr_2586] Constraints on `LifeCyclePeriod` ⌈The attributes `date`, `arReleaseVersion`, `productRelease` in `LifeCyclePeriod` are mutually exclusive.⌋

[constr_2587] No `System` in `AnyInstanceRef`

Imposition time: CP: `IT_InstRef`

⌈In consequence of [TPS_GST_00387] `System` shall not be `contextElement` nor `target` of an `AnyInstanceRef`. Otherwise `atpBase` would not be determined.⌋

[constr_2594] Cyclic value assignments to `SwSystemconst` is not allowed

Imposition time: CP: `IT_Variant`

⌈It is explicitly forbidden to assign values to `SwSystemconst` which in turn depend directly or indirectly on this value assignment.⌋

[constr_2595] Footnotes should not be nested ⌈Note that even if supported by the meta-model, footnotes shall not be nested. Nested footnotes might lead to problems with the processing of the footnote link. In other words `LParagraph` shall not be aggregated with role `ft` within a `LParagraph` which already has the role `ft`.⌋

[constr_2596] Used colors of attributes color and bgcolor ⌈The used colors of the attributes color and bgcolor shall base on the 6 digits RGB hex-code following

```
|#([a-fA-F0-9]{6})|
```

.⌋

[constr_2599] Maximum one VariationPoints in ≪`atpMixed`≫

Imposition time: CP: `IT_Variant`

⌈In case an ≪`atpMixed`≫ meta-class is aggregated as ≪`atpVariation`≫ there shall not be more than one `VariationPoint` and the `VariationPoint` shall be the last aggregated element.⌋

[constr_2601] Value of `AbstractEnumerationValueVariationPoint`

Imposition time: CP: `IT_Variant`

⌈The formula of an `AbstractEnumerationValueVariationPoint` shall evaluate to a value for which a mapping is defined in the `EnumerationMappingTable` which is referenced by the attributes `base` and `enumTable`.⌋

[constr_2606] Existence of `upperMultiplicityInfinite` and `upperMultiplicity` of `AbstractMultiplicityRestriction` is mutually exclu-

sive ⌈The existence of the elements upperMultiplicityInfinite and upperMultiplicity of AbstractMultiplicityRestriction shall be mutually exclusive.⌋

[constr_2607] lowerMultiplicity of AbstractMultiplicityRestriction shall be smaller or equal to upperMultiplicity ⌈lowerMultiplicity of AbstractMultiplicityRestriction shall be smaller or equal to upperMultiplicity.⌋

[constr_2626] atpTarget of InstanceRefs shall be consistent

Imposition time: CP: IT_InstRef

⌈The atpTarget of an instance ref shall either

- be an atpFeature owned by the atpType of the last atpContextElement or
- be an atpFeature owned by an AtpStructureElement owned by the atpType of the last atpContextElement.

⌋

[constr_2627] No reassigning of the same name within one LET Block ⌈Within one LET block one name shall be assigned to an value at most once.⌋

[constr_2628] Representation of xml.xsd.type=double data types ⌈All data types with xml.xsd.type=double shall comply with IEEE 754 and are limited to what can be expressed by a 64 bit binary representation.⌋

[constr_2629] Defined identity up to the root ⌈If an element in the M1 model aggregates splitable elements on deeper levels, it shall have a defined identity, i.e. the identifying attributes (e.g. shortName or shortLabel); see [TPS_GST_00047]; shall be set in the M1 model.⌋

[constr_2630] M1 elements with same identity but different type are not allowed ⌈Splitable M1 elements with the same identity but different type shall not exist, i.e. a merge conflict shall be reported.⌋

[constr_2631] Usage of value ANY for AnyServiceInstanceId ⌈The value of a given AnyServiceInstanceId shall not be set to ANY.⌋

[constr_2633] Existence of reference decorated with stereotype ≪isOfType≫ ⌈If a subclass of AtpPrototype defines a reference decorated with stereotype ≪isOfType≫ to a subclass of AtpType, then this reference shall always exist.⌋

[constr_2634] Ordered collections in Conditionals with property set patterns

Imposition time: CP: IT_Variant

⌈In the context of a class that is subject to the property set pattern, ordered collections shall not be split over different conditionals.⌋

[constr_2635] No custom values for `Collection.category` ⌈It is not allowed to define any custom or project-specific value of the attribute `Collection.category`.⌋

[constr_2636] No custom values for `Collection.elementRole` ⌈It is not allowed to define any custom or project-specific value of the attribute `Collection.elementRole`.⌋

[constr_2637] Limits of `PositiveUnlimitedInteger` ⌈For `PositiveUnlimitedInteger`, the min value can be down to 0 and the max value can be up to 18446744073709551615 (`0xffffffffffffffff`).⌋

[constr_2638] Variation points shall not exists in non-variant roles

Imposition time: CP: `IT_Variant`

⌈Variation Points shall not exist if the aggregation is not stereotyped ≪`atpVariation`≫ in the AUTOSAR meta model at any time in the workflow.⌋

[constr_2639] Restriction for the value of the first character in an `AnyVersionString` ⌈The first character of the value of any attribute typed by `AnyVersionString` shall only be in the range 1..9.⌋

[constr_2640] Restriction for the length of the value of an `AnyVersionString` ⌈The value of any attribute typed by `AnyVersionString` shall contain at least one character.⌋

[constr_2641] Mixed content elements cannot be split into different partial models ⌈The content of elements stereotyped with ≪`atpMixed`≫ or ≪`atpMixedString`≫ cannot be split, i.e. the mixed content element must be placed as a whole in one of the partial models.⌋

[constr_2643] `SwSystemconst` reference assignment to `SwSystemconstValue`

Imposition time: CP: `IT_Variant`

⌈If a `SwSystemconstValue` is applied to a `SwSystemconstantValueSet` a reference to `SwSystemconst` shall be given.⌋

[constr_2644] Assign the attribute `SwSystemconstValue.value` to `SwSystemconstValue`

Imposition time: CP: `IT_Variant`

⌈If a `SwSystemconstValue` is applied to a `SwSystemconstantValueSet` the attribute `SwSystemconstValue.value` shall be assigned to to `SwSystemconstValue`.⌋

[constr_9409] Existence of `AtpInstanceRef.atpTarget`

Imposition time: `IT_InstRef`

⌈For each subclass of `AtpInstanceRef`, the reference derived from the abstract role `atpTarget` shall exist.⌋

[constr_9410] Existence of `ConditionByFormula.bindingTime`

Imposition time: `IT_Variant`

⌈For each `ConditionByFormula`, attribute `bindingTime` shall exist.⌋

[constr_9411] Existence of `PostBuildVariantCondition.matchingCriterion`

Imposition time: `IT_Variant`

⌈For each `PostBuildVariantCondition`, attribute `matchingCriterion` shall exist.⌋

[constr_9412] Existence of `PostBuildVariantCondition.value`

Imposition time: `IT_Variant`

⌈For each `PostBuildVariantCondition`, attribute `value` shall exist.⌋

[constr_9413] Existence of `PostBuildVariantCriterion.compuMethod`

Imposition time: `IT_Variant`

⌈For each `PostBuildVariantCriterion`, attribute `compuMethod` shall exist.⌋

[constr_9414] Existence of `EvaluatedVariantSet.approvalStatus`

Imposition time: `IT_Variant`

⌈For each `EvaluatedVariantSet`, attribute `approvalStatus` shall exist.⌋

[constr_9415] Existence of `PostBuildVariantCriterionValue.variantCriterion`

Imposition time: `IT_Variant`

⌈For each `PostBuildVariantCriterionValue`, attribute `variantCriterion` shall exist.⌋

[constr_9416] Existence of `PostBuildVariantCriterionValue.value`

Imposition time: `IT_Variant`

⌈For each `PostBuildVariantCriterionValue`, attribute `value` shall exist.⌋

[constr_9417] Existence of `AnyInstanceRef.target`

Imposition time: `IT_InstRef`

⌈For each `AnyInstanceRef`, the reference in the role `target` shall exist.⌋

[constr_9418] Existence of `ShortNameFragment.fragment`

Imposition time: `IT_Name`

⌈For each `ShortNameFragment`, attribute `fragment` shall exist.⌋

[constr_9419] Existence of `ShortNameFragment.role`

Imposition time: `IT_Name`

⌈For each `ShortNameFragment`, attribute `role` shall exist.⌋

[constr_9420] Existence of `ReferenceBase.shortLabel`

Imposition time: `IT_RefBase`

⌈For each `ReferenceBase`, attribute `shortLabel` shall exist.⌋

[constr_10656] Value of a `SwSystemconst` used in the context of the *formula language for post-build configuration*

Imposition time: CP: `IT_Variant`

⌈If the combination of `SwSystemconst` and `SwSystemconstValue` is used in the context of the *formula language for post-build configuration*, then the effective value the involved `SwSystemconst` shall represent a **constant positive integer value** in the value range that can be represented by a 32-bit integer data type, i.e. 0 .. 4294967295.⌋

2.9 FO_TPS_StandardizationTemplate

[constr_2500] `PortInterface`s shall be of same kind ⌈Both objects (`PortInterface`s) referenced by a blueprint mapping for port interfaces (represented by `BlueprintMapping`) shall be of the same kind (e.g. both shall be `SenderReceiverInterface`s). In other words both interfaces shall be instances of the same meta class.⌋

[constr_2526] `PortInterface` need to be compatible to the blueprints ⌈`PortInterface` shall be compatible to their respective blueprints according to the compatibility rules.⌋

[constr_2527] Blueprints shall live in package of a proper category ⌈As explained in detail in the [12], model artifacts (in this case `PortPrototypeBlueprint` and incompletely specified `PortInterface`s) created for the purpose of becoming blueprints shall reside in an `ARPackage` of category `BLUEPRINT`.⌋

[constr_2528] `PortPrototype`s shall not refer to blueprints of a `PortInterface` ⌈A port `PortPrototype` shall not reference a `PortInterface` which lives in a package of category BLUEPRINT.⌋

[constr_2529] `PortPrototypeBlueprint`s and derived `PortPrototype`s shall reference proper `PortInterface`s ⌈A `PortPrototypeBlueprint` may reference a blueprint of `PortInterface`. According to [constr_2570], a system description shall not contain blueprints. Therefore the reference to the `PortInterface` may need to be rewritten when a `PortPrototype` is derived from the blueprint.

In this case the PortInterface referenced by the derived PortPrototype shall be compatible to the PortInterface (which is a blueprint) referenced by the PortPrototypeBlueprint.

According to [constr_2526] this can be ensured if the PortInterface referenced by the PortPrototypeBlueprint is the blueprint of the PortInterface referenced by the respective PortPrototype.⌋

[constr_2546] References in derived model elements ⌈Model elements derived from blueprints shall never refer to model elements that are blueprints.⌋

[constr_2553] shortName shall follow the pattern defined in the Blueprint ⌈The shortName respectively symbol of the derived objects shall follow the pattern defined in namePattern or blueprintValue of the blueprint according to [TPS_STDT_00086]⌋

[constr_2554] Derived objects shall match the blueprints ⌈Unless specified explicitly otherwise, the attributes of the blueprint shall appear in the derived objects. As an exception namePattern and blueprintValue may **not** be copied.⌋

[constr_2556] No Blueprint Motivated VariationPoints in AUTOSAR Descriptions ⌈AUTOSAR descriptions which are not blueprints shall not have blueprintCondition, formalBlueprintGenerator nor blueprintValue.⌋

[constr_2563] BswModuleDescription blueprints should not have a BswInternalBehavior ⌈A BswModuleDescription blueprint should not have a BswInternalBehavior since this is a matter of implementation and not subject to standardization. Exceptions might exist in vendor internal applications.⌋

[constr_2565] Traceable shall not be nested ⌈Due to the intended atomicity of requirements respectively specification items, Traceable shall not be nested.⌋

[constr_2568] SwComponentTypes shall be of same kind ⌈Both objects (SwComponentTypes) referenced by a blueprint mapping for port interfaces (represented by BlueprintMapping) shall be of the same kind (e.g. both shall be AtomicSwComponentTypes). In other words both components shall be instances of the same meta class.⌋

[constr_2569] Purely Blueprint Motivated VariationPoints ⌈VariationPoints with vh.latestBindingTime set to blueprintDerivationTime shall have only blueprintCondition or formalBlueprintGenerator respectively blueprintValue.⌋

[constr_2570] No Blueprints in system descriptions ⌈There shall be no blueprints in system descriptions. In consequence of this blueprint elements shall be referenced only from blueprints and AtpBlueprintMappings. Due to ≪atpUriDef≫, the references from AtpBlueprintMapping do not need to be resolved in system descriptions.⌋

[constr_2571] Outgoing references from Blueprints ⌈Note that outgoing references from Blueprints are basically not limited. Practically, references to objects living in a package of category EXAMPLE should not occur.⌋

[constr_2589] In VFB Timing Blueprint `TDEventVfbPort` shall reference `PortPrototypeBlueprint` ⌈In a VFB Timing Blueprint `TDEventVfbPort` shall reference `PortPrototypeBlueprint`. In other words, a VFB Timing Description Event specified in a VFB Timing Blueprint shall always reference a Port Prototype Blueprint.⌋

[constr_2590] One BlueprintPolicy is allowed ⌈For each attribute of a blueprint, at most one `BlueprintPolicy` is allowed.⌋

[constr_2591] BlueprintPolicyNotModifiable ⌈If `BlueprintPolicyNotModifiable` is assigned to an attribute, then during blueprinting it is not allowed to modify the value of the attribute and all its contained content.⌋

[constr_2592] No BlueprintPolicy ⌈If no `BlueprintPolicy` is assigned to an attribute, then arbitrary modifications are allowed while deriving from the blueprint.⌋

[constr_2593] Expression for identifying the attribute a BlueprintPolicy relates to ⌈The expression language for identifying the related attribute of a `BlueprintPolicy` is a subset version of xpath, see [13]. For navigation over the model we use the names as they are used in XML.⌋

[constr_2625] Permitted `LifeCycleState` combinations in a requirement uptrace ⌈

	Trace to: `TraceableText.category`=REQUIREMENT_ITEM			
Trace from:	DRAFT	VALID	OBSOLETE	REMOVED
DRAFT	1	1		
VALID	x	1		
OBSOLETE	1	1	1	
REMOVED	1	1	1	1
Legend:				
x) A "not applicable" requirement - as per [TPS_STDT_00056] with `LifeCycleState`==VALID may uptrace to `LifeCycleState`==DRAFT 1) Permitted				

⌋

[constr_6925] Existence of `PortPrototypeBlueprint.interface`

Imposition time: `IT_BlueprintCreation`

⌈For each `PortPrototypeBlueprint`, the reference in the role `interface` shall exist⌋

[constr_6933] `BlueprintMapping.derivedObject` shall not refer to a subclass of `AtpBlueprint` ⌈`BlueprintMapping.derivedObject` shall not refer to

a subclass of AtpBlueprint if the role blueprint refers to a PortPrototype-Blueprint.⌋

A Mentioned Class Tables

Class	**ARPackage**			
Note	AUTOSAR package, allowing to create top level packages to structure the contained AUElements. 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	*ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ARPackage.arPackage, AUTOSAR.arPackage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.shortName, element.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. **Tags:** atp.Splitkey=referenceBase, referenceBase.shortLabel xml.sequenceOffset=10

Table A.1: ARPackage

Class	**AUTOSAR**			
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:** atp.Splitkey=adminData xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30

△

Class	AUTOSAR			
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Stereotypes:** atpStructuredComment **Tags:** atp.Splitkey=fileInfoComment 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:** atp.Splitkey=introduction xml.sequenceOffset=20

Table A.2: AUTOSAR

Class	AbsoluteTolerance			
Note	Maximum allowable deviation			
Base	*ARObject*, *TimeRangeTypeTolerance*			
Aggregated by	TimeRangeType.tolerance			
Attribute	Type	Mult.	Kind	Note
absolute	TimeValue	0..1	attr	Maximum allowable deviation in duration (in seconds)

Table A.3: AbsoluteTolerance

Class	***AbstractAccessPoint*** (abstract)			
Note	Abstract class indicating an access point from an `ExecutableEntity`.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AsynchronousServerCallResultPoint, ExternalTriggeringPointIdent, InternalTriggeringPoint, ModeAccess PointIdent, ModeSwitchPoint, ParameterAccess, *ServerCallPoint*, VariableAccess			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	Type	Mult.	Kind	Note
returnValue Provision	RteApiReturnValue ProvisionEnum	0..1	attr	This attribute controls the provision of return values for RTE APIs that correspond to the enclosing access point. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.4: AbstractAccessPoint

Class	«atpVariation» ***AbstractCanCluster*** (abstract)			
Note	Abstract class that is used to collect the common J1939 and CAN Cluster attributes.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Subclasses	CanCluster, J1939Cluster			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	«atpVariation» ***AbstractCanCluster*** (abstract)			
busOffRecovery	CanClusterBusOff Recovery	0..1	aggr	CAN bus off monitoring / recovery at system level. **Tags:** atp.Splitkey=busOffRecovery
canFdBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the controller in bits/s.
canXlBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the CAN XL controller in bits/s.

Table A.5: AbstractCanCluster

Class	***AbstractCanCommunicationControllerAttributes*** (abstract)			
Note	For the configuration of the CanController parameters two different approaches can be used: 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			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canControllerFd Attributes	CanControllerFd Configuration	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. **Tags:** atp.Splitkey=canControllerFdAttributes
canControllerFd Requirements	CanControllerFd Configuration Requirements	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. **Tags:** atp.Splitkey=canControllerFdRequirements
canControllerXl Attributes	CanControllerXl Configuration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanXL frame. If this element exists the controller supports CanXL frames and the ECU developer shall take these values for the configuration of the CanXL controller. **Tags:** atp.Splitkey=canControllerXlAttributes
canControllerXl Requirements	CanControllerXl Configuration Requirements	0..1	aggr	Additional CanXL ranges of the bit timing related configuration of a CanXL controller. If this element exists the controller supports CanXL frames and the ECU developer shall take these ranges as requirements for the configuration of the CanXL controller. **Tags:** atp.Splitkey=canControllerXlRequirements

Table A.6: AbstractCanCommunicationControllerAttributes

Class	«atpMixedString» ***AbstractEnumerationValueVariationPoint*** (abstract)
Note	This is an abstract EnumerationValueVariationPoint. It is introduced to support the case that additional attributes are required for particular purposes.
Base	*ARObject*, *AttributeValueVariationPoint*, *FormulaExpression*, *SwSystemconstDependentFormula*
Subclasses	
Aggregated by	VariationPointProxy.valueAccess

▽

△

Class	«atpMixedString» ***AbstractEnumerationValueVariationPoint*** (abstract)			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	Identifier	0..1	attr	This attribute reflects the base to be used in context of EnumerationMappingTable for this reference. **Tags:** xml.attribute=true
enumTable	Ref	0..1	attr	This represents the assigned enumeration table. **Tags:** xml.attribute=true

Table A.7: AbstractEnumerationValueVariationPoint

Class	***AbstractEthernetFrame*** (abstract)			
Note	Ethernet specific attributes to the Frame.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	GenericEthernetFrame, UserDefinedEthernetFrame			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.8: AbstractEthernetFrame

Class	***AbstractEvent*** (abstract)			
Note	This meta-class represents the abstract ability to model an event that can be taken to implement application software or basic software in AUTOSAR.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*BswEvent*, *RTEEvent*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
activation Reason Representation	ExecutableEntity ActivationReason	0..1	ref	If the activationReasonRepresentation is referenced from the enclosing AbstractEvent this shall be taken as an indication that the latter contributes to the activating vector of this ExecutableEntity that owns the referenced ExecutableEntityActivationReason. **Tags:** atp.Splitkey=activationReasonRepresentation

Table A.9: AbstractEvent

Class	***AbstractImplementationDataType*** (abstract)			
Note	This meta-class represents an abstract base class for different flavors of ImplementationDataType.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ImplementationDataType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.10: AbstractImplementationDataType

Class	***AbstractImplementationDataTypeElement*** (abstract)			
Note	This meta-class represents the ability to act as an abstract base class for specific derived meta-classes that support the modeling of `ImplementationDataType`s for a particular language binding.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ImplementationDataTypeElement			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.11: AbstractImplementationDataTypeElement

Class	***AbstractMacSecCommunication*** (abstract)			
Note	Abstract class for the definition of a Connectivity Association and bypass for MACsec.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *SecurityConnectivityAssociation*			
Subclasses	MacSecBypassedCommunication, MacSecConnectivityAssociation			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort	CouplingPort	*	ref	Reference to the CouplingPorts this MacSec communication definition is handling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPort, couplingPort.couplingPort, couplingPort.variationPoint.shortLabel vh.latestBindingTime=postBuild
vlan	EthernetPhysical Channel	*	ref	Reference to the VLANs this MacSec communication definition is handling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=vlan, vlan.ethernetPhysicalChannel, vlan.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.12: AbstractMacSecCommunication

Class	***AbstractMessageToSecureChannelMapping*** (abstract)			
Note	Abstract class for the definition of mapping FrameTriggerings and SoConIPduIdentifiers to a secure channel. This may also include bypass mapping, where explicitly no security shall be applied.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	MessageToBypassCanSecCommunicationMapping, MessageToCanSecSecureChannelMapping			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canFrame Triggering	CanFrameTriggering	*	ref	Reference to the mapped CanFrameTriggerings. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canFrameTriggering, canFrame Triggering.canFrameTriggering, canFrame Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	***AbstractMessageToSecureChannelMapping*** (abstract)			
soConIPdu Identifier	SoConIPduIdentifier	*	ref	Reference to the mapped SoConIPduIdentifiers. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=soConIPduIdentifier, soConIPduIdentifier.so ConIPduIdentifier, soConIPduIdentifier.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.13: AbstractMessageToSecureChannelMapping

Class	***AbstractMultiplicityRestriction*** (abstract)			
Note	Restriction that specifies the valid number of occurrences of an element in the current context.			
Base	*ARObject*			
Subclasses	*SdgAttribute*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerMultiplicity	PositiveInteger	0..1	attr	Specifies the minimal number of times an object shall occur. If this primitive attribute is not set, then the object is optional.
upperMultiplicity	PositiveInteger	0..1	attr	Specifies the maximum number of times an object may occur. If this primitive attribute is not set, then there is no limit with respect to the maximum occurrence.
upperMultiplicity Infinite	Boolean	0..1	attr	This explicitly specifies, that the upper multiplicity is NOT restricted. Note: The use of 'upperMultiplicityInfinite' and 'upperMultiplicity' is mutual exclusive.

Table A.14: AbstractMultiplicityRestriction

Class	***AbstractProvidedPortPrototype*** (abstract)			
Note	This abstract class provides the ability to become a provided PortPrototype.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Subclasses	PPortPrototype, PRPortPrototype			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
providedCom Spec	PPortComSpec	*	aggr	Provided communication attributes per interface element (data element or operation). **Tags:** atp.Splitkey=providedComSpec, providedCom Spec.dataElement, providedComSpec.getter, provided ComSpec.modeGroup, providedComSpec.operation, providedComSpec.parameter, providedComSpec.setter, providedComSpec.variable

Table A.15: AbstractProvidedPortPrototype

Class	***AbstractRequiredPortPrototype*** (abstract)			
Note	This abstract class provides the ability to become a required PortPrototype.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Subclasses	PRPortPrototype, RPortPrototype			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***AbstractRequiredPortPrototype*** (abstract)			
requiredCom Spec	RPortComSpec	*	aggr	Required communication attributes, one for each interface element. **Tags:** atp.Splitkey=requiredComSpec, requiredCom Spec.dataElement, requiredComSpec.getter, required ComSpec.modeGroup, requiredComSpec.operation, requiredComSpec.parameter, requiredComSpec.setter, requiredComSpec.shortLabel, requiredComSpec.variable

Table A.16: AbstractRequiredPortPrototype

Class	***AbstractRuleBasedValueSpecification*** (abstract)			
Note	This represents an abstract base class for all rule-based value specifications.			
Base	*ARObject*, *ValueSpecification*			
Subclasses	ApplicationRuleBasedValueSpecification, CompositeRuleBasedValueSpecification, NumericalRuleBased ValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.17: AbstractRuleBasedValueSpecification

Class	***AbstractServiceInstance*** (abstract)			
Note	Provided and Consumed Ethernet Service Instances that are available at the ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ConsumedServiceInstance, *DdsCpServiceInstance*, ProvidedServiceInstance			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
capability Record	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=capabilityRecord, capabilityRecord.variation Point.shortLabel 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.

▽

△

Class	***AbstractServiceInstance*** (abstract)			
method Activation RoutingGroup	PduActivationRouting Group	0..1	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=methodActivationRoutingGroup, methodActivationRoutingGroup.shortName, methodActivationRoutingGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.18: AbstractServiceInstance

Class	**AccessCount**			
Note	This meta-class provides one count value for a `AbstractAccessPoint`.			
Base	*ARObject*			
Aggregated by	AccessCountSet.accessCount			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessPoint	AbstractAccessPoint	0..1	ref	AbstractAccessPoint for which the count value is applicable. **Tags:** atp.Splitkey=accessPoint
value	PositiveInteger	0..1	attr	This attribute represents the number of determined accesses **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.19: AccessCount

Class	**AccessCountSet**			
Note	This meta-class provides a set of count values evaluated according to the rules of a specific countProfile.			
Base	*ARObject*			
Aggregated by	ResourceConsumption.accessCountSet			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessCount	AccessCount	*	aggr	Count value for a AbstractAccessPoint. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=accessCount, accessCount.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
countProfile	NameToken	0..1	attr	This attribute defines the name of the count profile used to determine the AccessCount.value numbers.

Table A.20: AccessCountSet

Class	**AdminData**			
Note	AdminData represents the ability to express administrative information and custom extensions 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 the following kinds of meta-data • 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 Beside that a custom extension of model-data is possible by • Special data			
Base	*ARObject*			
Aggregated by	AUTOSAR.adminData, *Describable*.adminData, *Identifiable*.adminData			
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
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. **Tags:** atp.Splitkey=sdg, sdg.sdgCaption.shortName 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 MultilanguagePlain Text. 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:** atp.Splitkey=usedLanguages xml.sequenceOffset=30

Table A.21: AdminData

Class	AgeConstraint			
Note	Constrains the `scope` by a `minimum` and `maximum` time boundary.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maximum	MultidimensionalTime	0..1	aggr	The received event referenced by `scope` should not exceed this upper bound. **Tags:** atp.Splitkey=maximum
minimum	MultidimensionalTime	0..1	aggr	The received event referenced by `scope` should not precede this lower bound. **Tags:** atp.Splitkey=minimum
scope	TimingDescriptionEvent	0..1	ref	`TimingDescriptionEvent` to be constrained. **Tags:** atp.Splitkey=scope

Table A.22: AgeConstraint

Class	AliasNameAssignment			
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*			
Aggregated by	AliasNameSet.aliasName			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
flatInstance	FlatInstanceDescriptor	0..1	ref	Assignment of a unique name to a flat representation. **Tags:** atp.Splitkey=flatInstance xml.sequenceOffset=60 This Attribute is only used by the AUTOSAR Classic Platform.
identifiable	Identifiable	0..1	ref	Assignment of a unique name to an Identifiable. **Tags:** atp.Splitkey=identifiable xml.sequenceOffset=50
label	MultilanguageLong Name	0..1	aggr	This represents an "Alias LongName". **Tags:** atp.Splitkey=label xml.sequenceOffset=20
shortLabel	String	0..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. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10

Table A.23: AliasNameAssignment

Class	AliasNameSet
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

▽

△

Class	**AliasNameSet**			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aliasName	AliasNameAssignment	*	aggr	AliasNames contained in the AliasNameSet. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=aliasName, aliasName.shortLabel, aliasName.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.24: AliasNameSet

Class	**AnalyzedExecutionTime**			
Note	AnalyzedExecutionTime provides an analytic method for specifying the best and worst case execution time.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bestCase ExecutionTime	MultidimensionalTime	0..1	aggr	The best case execution time (BCET) defines the minimum amount of time the related executable entity requires for its execution. **Tags:** atp.Splitkey=bestCaseExecutionTime
worstCase ExecutionTime	MultidimensionalTime	0..1	aggr	The worst case execution time (WCET) defines the maximum amount of time the related executable entity requires for its execution. **Tags:** atp.Splitkey=worstCaseExecutionTime

Table A.25: AnalyzedExecutionTime

Class	**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	*ARObject*, *AtpInstanceRef*			
Aggregated by	ApmcInstanceReferenceValue.value, ApmcUpstreamDocInstanceReferenceValue.value, ApmcUriInstanceReferenceValue.value, Collection.collectedInstance, Collection.sourceInstance, DocumentationContext.feature, EcucInstanceReferenceValue.value, FlatInstanceDescriptor.ecuExtractReference, FlatInstanceDescriptor.upstreamReference, RptContainer.byPassPoint, RptHook.rptArHook, SecurityEventReportInstanceValue.object, ViewMap.firstElementInstance, ViewMap.secondElementInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	AtpClassifier	0..1	ref	This is the base from which navigation path begins. **Stereotypes:** atpDerived
contextElement (ordered)	AtpPrototype	*	ref	This is one step in the navigation path specified by the instance ref.
target	AtpFeature	0..1	ref	This is the target of the instance ref.

Table A.26: AnyInstanceRef

Primitive	**AnyServiceInstanceId**
Note	This is a positive integer or the literal ALL (the value ANY is technically supported but deprecated) which can be denoted in decimal, octal and hexadecimal. The value is between 0 and 65535. **Tags:** xml.xsd.customType=ANY-SERVICE-INSTANCE-ID xml.xsd.pattern=[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[0-7]*|0[bB][0-1]+|ANY|ALL xml.xsd.type=string

Table A.27: AnyServiceInstanceId

Primitive	**AnyVersionString**
Note	**Tags:** xml.xsd.customType=ANY-VERSION-STRING xml.xsd.pattern=[0-9]+|ANY xml.xsd.type=string

Table A.28: AnyVersionString

Class	**AppOsTaskProxyToEcuTaskProxyMapping**			
Note	This meta-class is used to map an OsTaskProxy that was created in the context of a SwComponent to an OsTaskProxy that was created in the context of an Ecu. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.appOsTaskProxyToEcuTaskProxyMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
appTaskProxy	OsTaskProxy	0..1	ref	Reference to an OsTaskProxy that is created in the context of a SwComponent. **Tags:** atp.Splitkey=appTaskProxy
ecuTaskProxy	OsTaskProxy	0..1	ref	Reference to an OsTaskProxy that is created in the context of an EcuInstance. **Tags:** atp.Splitkey=ecuTaskProxy
offset	Integer	0..1	attr	This attribute is used to describe the position of the app TaskProxy in an ecuTaskProxy as a relative value, i.e. the values show only the relative position of the appTask Proxy in the ecuTaskProxy.

Table A.29: AppOsTaskProxyToEcuTaskProxyMapping

Class	**ApplicationArrayDataType**			
Note	An application data type which is an array, each element is of the same application data type. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, *ARObject*, *ApplicationCompositeDataType*, *ApplicationDataType*, *AtpBlueprint*, *Atp Classifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.

▽

△

Class	**ApplicationArrayDataType**			
element	ApplicationArray Element	0..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. **Tags:** atp.Splitkey=element, element.shortName

Table A.30: ApplicationArrayDataType

Class	**ApplicationArrayElement**			
Note	Describes the properties of the elements of an application array data type.			
Base	*ARObject*, *ApplicationCompositeElementDataPrototype*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationArrayDataType.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled.
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls how the information about the array size shall be interpreted.
indexDataType	ApplicationPrimitive DataType	0..1	ref	This reference can be taken to assign a `CompuMethod` of category `TEXTTABLE` to the array. The texttable entries associate a textual value to an index number such that the element with that index number is represented by a symbolic name. **Tags:** atp.Splitkey=indexDataType
maxNumberOf Elements	PositiveInteger	0..1	attr	The maximum number of elements that the array can contain. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.31: ApplicationArrayElement

Class	***ApplicationCompositeDataType*** (abstract)			
Note	Abstract base class for all application data types composed of other data types.			
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ApplicationArrayDataType, ApplicationRecordDataType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.32: ApplicationCompositeDataType

Class	**ApplicationCompositeDataTypeSubElementRef**			
Note	This meta-class represents the specialization of `SubElementMapping` with respect to `ApplicationCompositeDataType`s.			
Base	*ARObject*, *SubElementRef*			
Aggregated by	SubElementMapping.firstElement, SubElementMapping.secondElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ApplicationCompositeDataTypeSubElementRef**			
application Composite Element	ApplicationComposite ElementDataPrototype	0..1	iref	This represents the referenced ApplicationComposite DataPrototype. **Tags:** atp.Splitkey=applicationCompositeElement, applicationCompositeElement.rootDataPrototype, applicationCompositeElement.contextDataPrototype, applicationCompositeElement.targetDataPrototype **InstanceRef implemented by:** ApplicationComposite ElementInPortInterfaceInstanceRef

Table A.33: ApplicationCompositeDataTypeSubElementRef

Class	***ApplicationCompositeElementDataPrototype*** (abstract)			
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			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	ApplicationDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table A.34: ApplicationCompositeElementDataPrototype

Class	**ApplicationCompositeElementInPortInterfaceInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ApplicationCompositeDataTypeSubElementRef.applicationCompositeElement, CompositeNetwork Representation.leafElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	DataInterface	0..1	ref	This represents the SenderReceiverInterface that acts as the base in this InstanceRef definition **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This represents a context ApplicationCompositeData Prototype **Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	This refers to the dataPrototype which is typed by the ApplicationDatatype in which which the target can be found. **Tags:** xml.sequenceOffset=15
targetData Prototype	ApplicationComposite ElementDataPrototype	0..1	ref	This represents the referenced ApplicationComposite DataPrototype. **Tags:** xml.sequenceOffset=30

Table A.35: ApplicationCompositeElementInPortInterfaceInstanceRef

Class	**ApplicationDataType** (abstract)			
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, endianess, etc. It should be possible to model the application level aspects of a VFB system by using `ApplicationDataType`s only.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*ApplicationCompositeDataType*, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.36: ApplicationDataType

Class	**ApplicationEndpoint**			
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*			
Aggregated by	SocketAddress.applicationEndpoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxNumberOf Connections	PositiveInteger	0..1	attr	This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.
network Endpoint	NetworkEndpoint	0..1	ref	Reference to the network address. **Tags:** atp.Splitkey=networkEndpoint
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
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.Splitkey=tlsCryptoMapping
tpConfiguration	TransportProtocol Configuration	0..1	aggr	Configuration of the used transport protocol. **Tags:** atp.Splitkey=tpConfiguration

Table A.37: ApplicationEndpoint

Class	**ApplicationEntry**			
Note	Schedule table entry for application messages. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frameTriggering	LinFrameTriggering	0..1	ref	Specifies the LinFrame that will be transmitted in this frame slot. **Tags:** atp.Splitkey=frameTriggering

Table A.38: ApplicationEntry

Class	ApplicationError			
Note	This is a user-defined error that is associated with an element of an AUTOSAR interface. It is specific for the particular functionality or service provided by the AUTOSAR software component.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ClientServerInterface.possibleError			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
errorCode	Integer	0..1	attr	The RTE generator is forced to assign this value to the corresponding error symbol. Note that for error codes certain ranges are predefined (see RTE specification).

Table A.39: ApplicationError

Class	ApplicationPartition			
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 This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.40: ApplicationPartition

Class	ApplicationPartitionToEcuPartitionMapping			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.applicationPartitionToEcuPartitionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Partition	ApplicationPartition	*	ref	Reference to ApplicationPartitions that are mapped to an EcuPartition. **Tags:** atp.Splitkey=applicationPartition
ecuPartition	EcuPartition	0..1	ref	Reference to EcuPartition to which the Application Partitions are assigned. **Tags:** atp.Splitkey=ecuPartition

Table A.41: ApplicationPartitionToEcuPartitionMapping

Class	ApplicationPrimitiveDataType			
Note	A primitive data type defines a set of allowed values. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.42: ApplicationPrimitiveDataType

Class	**ApplicationRecordDataType**			
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*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
element (ordered)	ApplicationRecordElement	*	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 A.43: ApplicationRecordDataType

Class	**ApplicationRecordElement**			
Note	Describes the properties of one particular element of an application record data type.			
Base	*ARObject*, *ApplicationCompositeElementDataPrototype*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationRecordDataType.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
isOptional	Boolean	0..1	attr	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. 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.

Table A.44: ApplicationRecordElement

Class	**ApplicationRuleBasedValueSpecification**
Note	This meta-class represents rule based values for DataPrototypes typed by ApplicationDataTypes (ApplicationArrayDataType or a compound ApplicationPrimitiveDataType which also boils down to an array-nature).
Base	*ARObject*, *AbstractRuleBasedValueSpecification*, *CompositeRuleBasedValueArgument*, *ValueSpecification*
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.compoundPrimitiveArgument, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableDataPrototype.initValue

▽

△

Class	**ApplicationRuleBasedValueSpecification**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	Identifier	0..1	attr	This represents the category of the RuleBasedValue Specification **Tags:** xml.sequenceOffset=-20
swAxisCont (ordered)	RuleBasedAxisCont	*	aggr	This represents the axis values of a Compound Primitive Data Type (curve or map). The first swAxisCont describes the x-axis, the second sw AxisCont describes the y-axis, the third swAxisCont describes the z-axis. In addition to this, the axis can be denoted in swAxisIndex.
swValueCont	RuleBasedValueCont	0..1	aggr	This represents the values of an array or Compound Primitive Data Type. **Tags:** atp.Splitkey=swValueCont

Table A.45: ApplicationRuleBasedValueSpecification

Class	**ApplicationSwComponentType**			
Note	The `ApplicationSwComponentType` is used to represent the application software. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.46: ApplicationSwComponentType

Class	**ApplicationValueSpecification**			
Note	This meta-class represents values for DataPrototypes typed by ApplicationDataTypes (this includes in particular compound primitives). For further details refer to ASAM CDF 2.0. This meta-class corresponds to some extent with SW-INSTANCE in ASAM CDF 2.0.			
Base	*ARObject*, *CompositeRuleBasedValueArgument*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.compoundPrimitive Argument, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvData Condition.compareValue, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdData Condition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitution Value, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.rom BlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideCom Spec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInit Value.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, Sw DataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	Identifier	0..1	attr	Specifies to which category of ApplicationDataType this ApplicationValueSpecification can be applied (e.g. as an initial value), thus imposing constraints on the structure and semantics of the contained values, see [constr_1006] and [constr_1519].

▽

△

Class	**ApplicationValueSpecification**			
swAxisCont (ordered)	SwAxisCont	*	aggr	This represents the axis values of a Compound Primitive Data Type (curve or map). The first swAxisCont describes the x-axis, the second sw AxisCont describes the y-axis, the third swAxisCont describes the z-axis. In addition to this, the axis can be denoted in swAxisIndex.
swValueCont	SwValueCont	0..1	aggr	This represents the values of a Compound Primitive Data Type. **Tags:** atp.Splitkey=swValueCont

Table A.47: ApplicationValueSpecification

Class	**ArParameterInImplementationDataInstanceRef**			
Note	This class represents the ability to navigate into an element inside of an ParameterDataPrototype typed by an ImplementationDatatype. Note that it shall not be used if the target is the ParameterDataPrototype itself (e.g. if the target is a primitive data type). Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataType Element (intentionally) isn't derived from AtpPrototype.			
Base	*ARObject*			
Aggregated by	ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextData Prototype (ordered)	AbstractImplementation DataTypeElement	*	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. **Stereotypes:** atpIdentityContributor
portPrototype	PortPrototype	0..1	ref	This reference points to the PortPrototype providing/ receiving the root of the parameter. **Stereotypes:** atpIdentityContributor
rootParameter DataPrototype	ParameterData Prototype	0..1	ref	This refers to the ParameterDataPrototype typed by the implementationDataType in which the target can be found. **Stereotypes:** atpIdentityContributor
targetData Prototype	AbstractImplementation DataTypeElement	0..1	ref	This reference points to the target ImplementationData TypeElement. **Stereotypes:** atpIdentityContributor

Table A.48: ArParameterInImplementationDataInstanceRef

Class	**ArVariableInImplementationDataInstanceRef**
Note	This class represents the ability to navigate into a data element inside of an VariableDataPrototype which is typed by an ImplementationDatatype. Note that it shall not be used if the target is the VariableDataPrototype itself (e.g. if its a primitive). Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataType Element isn't derived from AtpPrototype.
Base	*ARObject*
Aggregated by	AutosarVariableRef.autosarVariableInImplDatatype, ImplementationDataTypeSubElement Ref.implementationDataTypeElement

▽

△

Class	**ArVariableInImplementationDataInstanceRef**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextData Prototype (ordered)	AbstractImplementation DataTypeElement	*	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. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=30
portPrototype	PortPrototype	0..1	ref	This is the port providing/receiving the root of the variable **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10
rootVariable DataPrototype	VariableDataPrototype	0..1	ref	This refers to the VariableDataPrototype typed by the ImplementationDatatype in which the target can be found. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=20
targetData Prototype	AbstractImplementation DataTypeElement	0..1	ref	This reference points to the target ImplementationData TypeElement. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=40

Table A.49: ArVariableInImplementationDataInstanceRef

Class	**ArbitraryEventTriggering**			
Note	Describes that an event occurs occasionally, singularly, irregularly or randomly. The primary purpose of this event triggering is to abstract event occurrences captured by data acquisition tools (background debugger, trace analyzer, etc.) during system runtime.			
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Timing Constraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
confidence Interval	ConfidenceInterval	*	aggr	List of confidence intervals. **Tags:** atp.Splitkey=confidenceInterval xml.sequenceOffset=30
maximum Distance	MultidimensionalTime	*	aggr	The nth array element describes the maximum distance that can be observed for a sample of n+1 event occurrences. This is an array with an identical number of elements as for the minimumDistance. **Tags:** atp.Splitkey=maximumDistance xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=20 xml.typeElement=false

▽

△

Class	**ArbitraryEventTriggering**			
minimum Distance	MultidimensionalTime	*	aggr	The nth array element describes the minimum distance that can be observed for a sample of n+1 event occurrences. This is an array with an identical number of elements as for the maximumDistance. **Tags:** atp.Splitkey=minimumDistance xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=10 xml.typeElement=false

Table A.50: ArbitraryEventTriggering

Class	**ArgumentDataPrototype**			
Note	An argument of an operation, carries direction and implementation information.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ClientServerOperation.argument			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
direction	ArgumentDirectionEnum	0..1	attr	This attribute specifies the direction of the argument.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	0..1	attr	This defines how the argument type of the servers `RunnableEntity` is implemented. 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.

Table A.51: ArgumentDataPrototype

Enumeration	**ArgumentDirectionEnum**
Note	Use cases: • 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.
Aggregated by	ArgumentDataPrototype.direction, DiagnosticSovdAccessArgument.direction, RunnableEntityArgument.direction, SwServiceArg.direction
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 A.52: ArgumentDirectionEnum

Enumeration	**ArraySizeHandlingEnum**
Note	This enumeration defines different ways to handle the sizes of variable size arrays.
Aggregated by	ApplicationArrayElement.arraySizeHandling, ImplementationDataTypeElement.arraySizeHandling
Literal	***Description***
allIndicesDifferent ArraySize	All elements of the variable size array may have different sizes. **Tags:** atp.EnumerationLiteralIndex=0
allIndicesSame ArraySize	All elements of the variable size array have the same size. **Tags:** atp.EnumerationLiteralIndex=1
inheritedFromArray ElementTypeSize	The size of all dimensions of the variable size array is determined by the size of the contained array element. **Tags:** atp.EnumerationLiteralIndex=2

Table A.53: ArraySizeHandlingEnum

Enumeration	**ArraySizeSemanticsEnum**
Note	This enumeration provides options for how the number of elements in an array object shall be interpreted.
Aggregated by	ApplicationArrayElement.arraySizeSemantics, DiagnosticDataElement.arraySizeSemantics, ImplementationDataTypeElement.arraySizeSemantics, SwTextProps.arraySizeSemantics
Literal	***Description***
fixedSize	This means that the array object will always have a fixed number of elements. **Tags:** atp.EnumerationLiteralIndex=0
variableSize	This implies that the actual number of elements in the array object might vary at run-time. The configured size represents the maximum number of elements in the array object. **Tags:** atp.EnumerationLiteralIndex=1

Table A.54: ArraySizeSemanticsEnum

Class	**ArrayValueSpecification**			
Note	Specifies the values for an array.			
Base	*ARObject*, *CompositeValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.argument, Constant Specification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compare Value, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compare Value, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
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

▽

△

Class	ArrayValueSpecification			
intendedPartial Initialization Count	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 A.55: ArrayValueSpecification

Class	AssemblySwConnector			
Note	`AssemblySwConnector`s are exclusively used to connect `SwComponentPrototype`s in the context of a `CompositionSwComponentType`.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port. **Tags:** atp.Splitkey=provider, provider.contextComponent, provider.targetPPort **InstanceRef implemented by:** PPortInComposition InstanceRef
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port. **Tags:** atp.Splitkey=requester, requester.context Component, requester.targetRPort **InstanceRef implemented by:** RPortInComposition InstanceRef

Table A.56: AssemblySwConnector

Class	AssignFrameId			
Note	Schedule entry for an Assign Frame Id master request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
assignedFrame Triggering	LinFrameTriggering	0..1	ref	The frame whose identifier is set by this assignment. **Tags:** atp.Splitkey=assignedFrameTriggering

Table A.57: AssignFrameId

Class	AssignFrameIdRange			
Note	AssignFrameIdRange generates an assign frame PID range request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**AssignFrameIdRange**			
framePid	FramePid	0..4	aggr	Optional assignment of frame_PID values that are included in the request. The frame_PIDs are ordered. **Tags:** atp.Splitkey=framePid
startIndex	Integer	0..1	attr	The startIndex sets the index to the first frame to assign a PID.

Table A.58: AssignFrameIdRange

Class	**AssignNad**			
Note	Schedule entry for an Assign NAD master request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
newNad	Integer	0..1	attr	The newly assigned NAD value.

Table A.59: AssignNad

Class	**AsynchronousServerCallPoint**			
Note	An AsynchronousServerCallPoint is used for asynchronous invocation of a ClientServerOperation. IMPORTANT: a ServerCallPoint cannot be used concurrently. Once the client RunnableEntity has made the invocation, the ServerCallPoint cannot be used until the call returns (or an error occurs!) at which point the ServerCallPoint becomes available again. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServerCallPoint*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.60: AsynchronousServerCallPoint

Class	**AsynchronousServerCallResultPoint**			
Note	If a `RunnableEntity` owns an `AsynchronousServerCallResultPoint` it is entitled to get the result of the referenced `AsynchronousServerCallPoint`. If it is associated with an `AsynchronousServerCallReturnsEvent`, this `RTEEvent` notifies the completion of the required `ClientServerOperation` or a timeout. The occurrence of this event can either unblock a `WaitPoint` or can lead to the invocation of a `RunnableEntity`. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.asynchronousServerCallResultPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
asynchronous ServerCallPoint	AsynchronousServer CallPoint	0..1	ref	The referenced Asynchronous Server Call Point defines the asynchronous server call from which the results are returned. **Tags:** atp.Splitkey=asynchronousServerCallPoint

Table A.61: AsynchronousServerCallResultPoint

Class	AsynchronousServerCallReturnsEvent			
Note	This event is raised when an asynchronous server call is finished.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	AsynchronousServerCallResultPoint	0..1	ref	The referenced AsynchronousServerCallResultPoint raises this AsynchronousServerCallReturnsEvent when the asynchronous server call returns. **Tags:** atp.Splitkey=eventSource This Attribute is only used by the AUTOSAR Classic Platform.

Table A.62: AsynchronousServerCallReturnsEvent

Class	***AtomicSwComponentType*** (abstract)			
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*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Subclasses	ApplicationSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponentType, NvBlockSwComponentType, SensorActuatorSwComponentType, ServiceProxySwComponentType, ServiceSwComponentType			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
internalBehavior	SwcInternalBehavior	0..1	aggr	The `SwcInternalBehavior`s owned by an `AtomicSwComponentType` can be located in a different physical file. Therefore the aggregation is <<atp Splitable>>. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalBehavior, internalBehavior.shortName, internalBehavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the `SymbolProps` for the `AtomicSwComponentType`. **Tags:** atp.Splitkey=symbolProps, symbolProps.shortName

Table A.63: AtomicSwComponentType

Class	***AtpBlueprint*** (abstract)			
Note	This meta-class represents the ability to act as a Blueprint. As this class is an abstract one, particular blueprint meta-classes inherit from this one.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ARPackage, *AbstractImplementationDataType*, AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, *ApplicationDataType*, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, *BuildActionEntity*, BuildActionEnvironment, BuildActionManifest, CompuMethod, ConsistencyNeeds, DataConstr, DataTypeMappingSet, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleDef, FlatMap, ImpositionTime, ImpositionTimeDefinitionGroup, KeywordSet, LifeCycleState, LifeCycleStateDefinitionGroup, ModeDeclarationGroup, *PortInterface*, *PortInterfaceMapping*, PortInterfaceMappingSet, PortPrototypeBlueprint, SecurityEventContextDataElement, SwAddrMethod, SwBaseType, *SwComponentType*, VfbTiming			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	***AtpBlueprint*** (abstract)			
blueprintPolicy	BlueprintPolicy	*	aggr	This role indicates whether the blueprintable element will be modifiable or not modifiable. **Tags:** atp.Splitkey=blueprintPolicy, blueprint Policy.attributeName

Table A.64: AtpBlueprint

Class	***AtpBlueprintMapping*** (abstract)			
Note	This meta-class represents the ability to express a particular mapping between a blueprint and an element derived from this blueprint. Particular mappings are defined by specializations of this meta-class.			
Base	*ARObject*			
Subclasses	BlueprintMapping			
Aggregated by	BlueprintMappingSet.blueprintMap			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atpBlueprint	AtpBlueprint	0..1	ref	This represents the blueprint. **Stereotypes:** atpAbstract; atpUriDef **Tags:** xml.sequenceOffset=50
atpBlueprinted Element	Identifiable	0..1	ref	This represents the bluprinted elements which shall be mapped to the blueprint. **Stereotypes:** atpAbstract **Tags:** xml.sequenceOffset=60

Table A.65: AtpBlueprintMapping

Class	***AtpClassifier*** (abstract)			
Note	A classifier classifies M0 instances according to their features. Or: a classifier is something that has instances - an M1 classifier has M0 instances.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AtpStructureElement*, *AtpType*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atpFeature	AtpFeature	*	aggr	This is a feature of the classifier. **Stereotypes:** atpDerived

Table A.66: AtpClassifier

Class	***AtpFeature*** (abstract)			
Note	Features are properties via which a classifier classifies instances. Or: a classifier has features and every M0 instance of it will have those features.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AtpPrototype*, *AtpStructureElement*			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.67: AtpFeature

Class	***AtpInstanceRef*** (abstract)			
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, ComponentInComposition InstanceRef, ComponentInSystemInstanceRef, *DataPrototypeInPortInterfaceInstanceRef*, Data PrototypeInSystemInstanceRef, InnerDataPrototypeGroupInCompositionInstanceRef, InnerPortGroupIn CompositionInstanceRef, InnerRunnableEntityGroupInCompositionInstanceRef, InstanceEventIn CompositionInstanceRef, ModeDeclarationGroupPrototypeInSystemInstanceRef, *ModeGroupInAtomic SwcInstanceRef*, ModeInBswInstanceRef, ModeInBswModuleDescriptionInstanceRef, ModeInSwc InstanceRef, OperationArgumentInComponentInstanceRef, *OperationInAtomicSwcInstanceRef*, OperationInSystemInstanceRef, PModeInSystemInstanceRef, ParameterDataPrototypeInSystem InstanceRef, ParameterInAtomicSWCTypeInstanceRef, PortGroupInSystemInstanceRef, *PortIn CompositionTypeInstanceRef*, RModeInAtomicSwcInstanceRef, RteEventInCompositionInstanceRef, RteEventInEcuInstanceRef, RteEventInSystemInstanceRef, RunnableEntityInCompositionInstanceRef, SwcServiceDependencyInSystemInstanceRef, *TriggerInAtomicSwcInstanceRef*, TriggerInSystem InstanceRef, VariableAccessInEcuInstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstanceRef, VariableInAtomicSWCTypeInstanceRef, *VariableInAtomic SwcInstanceRef*, VariableInComponentInstanceRef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atpBase	AtpClassifier	0..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
atpTarget	AtpFeature	0..1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table A.68: AtpInstanceRef

Class	***AtpPrototype*** (abstract)			
Note	A prototype is a typed feature. A prototype in a classifier indicates that instances of that classifier will have a feature, and the structure of that feature is given by the its type. An instance of that type will play the role indicated by the feature in the owning classifier. A feature is not an instance but an indication of an instance-to-be.			
Base	*ARObject*, *AtpFeature*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*DataPrototype*, ModeDeclarationGroupPrototype, *PortPrototype*, RootSwCompositionPrototype, Sw ComponentPrototype			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atpType	AtpType	0..1	ref	This is the type of the feature. **Stereotypes:** atpAbstract

Table A.69: AtpPrototype

Class	***AtpStructureElement*** (abstract)
Note	A structure element is both a classifier and a feature. As a feature, its structure is given by the feature it owns as a classifier.
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	***AtpStructureElement*** (abstract)			
Subclasses	*AbstractAccessPoint*, *AbstractImplementationDataTypeElement*, AsynchronousServerCallResultPoint, BswModuleDescription, BulkNvDataDescriptor, ClientServerOperation, DataPrototypeGroup, *Ident Caption*, *InternalBehavior*, InternalTriggeringPoint, ModeDeclaration, ModeDeclarationMapping, Mode SwitchPoint, ModeTransition, NvBlockDescriptor, ParameterAccess, PerInstanceMemory, PortGroup, PortPrototypeBlueprint, *RTEEvent*, RunnableEntity, RunnableEntityGroup, *ServerCallPoint*, *Sw Connector*, SwcBswMapping, SwcServiceDependency, System, Trigger, VariableAccess			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.70: AtpStructureElement

Class	***AtpType*** (abstract)			
Note	A type is a classifier that may serve to type prototypes. It is a reusable classifier.			
Base	*ARObject*, *AtpClassifier*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AutosarDataType*, ModeDeclarationGroup, ModeDeclarationMappingSet, *PortInterface*, *SwComponent Type*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.71: AtpType

Class	«atpMixedString» ***AttributeValueVariationPoint*** (abstract)			
Note	This class represents the ability to derive the value of the Attribute from a system constant (by Sw SystemconstDependentFormula). It also provides a bindingTime.			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Subclasses	*AbstractEnumerationValueVariationPoint*, *AbstractNumericalVariationPoint*, BooleanValueVariationPoint, FloatValueVariationPoint, IntegerValueVariationPoint, PositiveIntegerValueVariationPoint, TimeValue ValueVariationPoint, UnlimitedIntegerValueVariationPoint			
Aggregated by	VariationPointProxy.valueAccess			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bindingTime	FullBindingTimeEnum	0..1	attr	This is the binding time in which the attribute value needs to be bound. If this attribute is missing, the attribute is not a variation point. In particular this means that It needs to be a single value according to the type specified in the pure model. It is an error if it is still a formula. **Tags:** xml.attribute=true
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** xml.attribute=true
sd	String	0..1	attr	This special data is provided to allow synchronization of Attribute value variation points with variant management systems. The usage is subject of agreement between the involved parties. **Tags:** xml.attribute=true

▽

△

Class	«atpMixedString» ***AttributeValueVariationPoint*** (abstract)			
shortLabel	PrimitiveIdentifier	0..1	attr	This allows to identify the variation point. It is also intended to allow RTE support for CompileTime Variation points. **Tags:** xml.attribute=true

Table A.72: AttributeValueVariationPoint

Class	***AutosarDataPrototype*** (abstract)			
Note	Base class for prototypical roles of an `AutosarDataType`.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ArgumentDataPrototype, ParameterDataPrototype, VariableDataPrototype			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	AutosarDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table A.73: AutosarDataPrototype

Class	***AutosarDataType*** (abstract)			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractImplementationDataType*, *ApplicationDataType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType. **Tags:** atp.Splitkey=swDataDefProps

Table A.74: AutosarDataType

Class	**AutosarOperationArgumentInstance**			
Note	This class represents a reference to an argument instance. This way it is possible to reference an argument instance in the occurrence expression formula. The argument instance can target to one of the following arguments: • a whole argument used in an operation of a `PortPrototype` typed by a `ClientServerInterface` • an element inside of a composite argument used in an operation of a `PortPrototype` typed by a `ClientServerInterface`			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TDEventOccurrenceExpression.argument, TimingExtensionResource.timingArgument			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**AutosarOperationArgumentInstance**			
operation Argument Instance	DataPrototype	0..1	iref	This is the reference to the instanceRef definition. **Tags:** atp.Splitkey=operationArgumentInstance, operationArgumentInstance.base, operationArgument Instance.contextComponent, operationArgument Instance.contextPortPrototype, operationArgument Instance.contextOperation, operationArgument Instance.rootArgumentDataPrototype, operationArgument Instance.contextDataPrototype, operationArgument Instance.targetDataPrototype **InstanceRef implemented by:** OperationArgumentIn ComponentInstanceRef

Table A.75: AutosarOperationArgumentInstance

Class	**AutosarParameterRef**			
Note	This class represents a reference to a parameter within AUTOSAR which can be one of the following use cases: localParameter: • localParameter which is used as whole (e.g. sharedAxis for curve) autosarVariable: • a parameter provided via `PortPrototype` which is used as whole (e.g. parameterAccess) • an element inside of a composite local parameter typed by `ApplicationDataType` (e.g. sharedAxis for a curve) • an element inside of a composite parameter provided via Port and typed by `ApplicationDataType` (e.g. sharedAxis for a curve) autosarParameterInImplDatatype: • an element inside of a composite local parameter typed by ImplementationDatatype • an element inside of a composite parameter provided via PortPrototype and typed by Implementation Datatype			
Base	*ARObject*			
Aggregated by	InstantiationDataDefProps.parameterInstance, ParameterAccess.accessedParameter, RoleBasedData Assignment.usedParameterElement, SwCalprmRefProxy.arParameter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
autosar Parameter	DataPrototype	0..1	iref	This instance reference is used if the calibration parameter is either imported via a port or is part of a composite data structure. **Tags:** atp.Splitkey=autosarParameter, autosar Parameter.portPrototype, autosarParameter.root ParameterDataPrototype, autosarParameter.contextData Prototype, autosarParameter.targetDataPrototype **InstanceRef implemented by:** ParameterInAtomic SWCTypeInstanceRef

▽

△

Class	**AutosarParameterRef**			
localParameter	DataPrototype	0..1	ref	In the majority of cases this reference goes to Parameter DataPrototypes rather than VariableDataPrototypes. Pointing the reference to a VariableDataPrototype is limited to special use cases, e.g. if the AutosarParameter Ref is used in the context of an SwAxisGrouped. This reference is used if the arParameter is local to the current component. Of course, it would technically also be feasible to use an InstanceRef for this case. However, the InstanceRef would not have a contextElement (because the current instance is the context). Hence, the local instance is a special case which may provide further optimization. Therefore an explicit reference is provided for this case. **Tags:** atp.Splitkey=localParameter

Table A.76: AutosarParameterRef

Class	**AutosarVariableInstance**			
Note	Represents a reference to a variable instance. The target of the variable instance can be either: • a variable provided via a `PortPrototype` • an element inside of a composite variable (e.g. struct) provided via a `PortPrototype`			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TDEventOccurrenceExpression.variable, TimingExtensionResource.timingVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
variableInstance	DataPrototype	0..1	iref	This is the reference to the instanceRef definition. **Tags:** atp.Splitkey=variableInstance, variable Instance.contextComponent, variableInstance.contextPort Prototype, variableInstance.rootVariableDataPrototype, variableInstance.contextDataPrototype, variable Instance.targetDataProtoype **InstanceRef implemented by:** VariableInComponent InstanceRef

Table A.77: AutosarVariableInstance

Class	AutosarVariableRef			
Note	This class represents a reference to a variable within AUTOSAR which can be one of the following use cases: localVariable: • localVariable which is used as whole (e.g. InterRunnableVariable, inputValue for curve) autosarVariable: • a variable provided via Port which is used as whole (e.g. dataAccessPoints) • an element inside of a composite local variable typed by `ApplicationDataType` (e.g. inputValue for a curve) • an element inside of a composite variable provided via `PortPrototype` and typed by `ApplicationDataType` (e.g. inputValue for a curve) autosarVariableInImplDatatype: • an element inside of a composite local variable typed by `ImplementationDataType` (e.g. nvram Data mapping) • an element inside of a composite variable provided via Port and typed by `ImplementationDataType` (e.g. inputValue for a curve)			
Base	*ARObject*			
Aggregated by	InstantiationDataDefProps.variableInstance, NvBlockDataMapping.nvRamBlockElement, NvBlockDataMapping.readNvData, NvBlockDataMapping.writtenNvData, NvBlockDataMapping.writtenReadNvData, RoleBasedDataAssignment.usedDataElement, SwVariableRefProxy.autosarVariable, VariableAccess.accessedVariable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autosarVariable	DataPrototype	0..1	iref	This references a variable which is provided by a port and/or which is part of a CompositeDataType. **Tags:** atp.Splitkey=autosarVariable, autosarVariable.portPrototype, autosarVariable.rootVariableDataPrototype, autosarVariable.contextDataPrototype, autosarVariable.targetDataPrototype **InstanceRef implemented by:** VariableInAtomicSWCTypeInstanceRef
autosarVariableInImplDatatype	ArVariableInImplementationDataInstanceRef	0..1	aggr	This is used if the target variable is inside of variableDataPrototype typed by an ImplementationDataType. **Tags:** atp.Splitkey=autosarVariableInImplDatatype, autosarVariableInImplDatatype.portPrototype, autosarVariableInImplDatatype.rootVariableDataPrototype, autosarVariableInImplDatatype.contextDataPrototype, autosarVariableInImplDatatype.targetDataPrototype
localVariable	VariableDataPrototype	0..1	ref	This reference is used if the variable is local to the current component. It would also be possible to use the instance refence here. Such an instance ref would not have a contextElement, since the current instance is the context. But the local instance is a special case which may provide further optimization. Therefore an explicit reference is provided for this case. **Tags:** atp.Splitkey=localVariable

Table A.78: AutosarVariableRef

Class	**BaseType** (abstract)			
Note	This abstract meta-class represents the ability to specify a platform dependent base type.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	SwBaseType			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	***BaseType*** (abstract)			
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 A.79: BaseType

Class	**BaseTypeDirectDefinition**			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Aggregated by	*BaseType*.baseTypeDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseType Encoding	BaseTypeEncoding String	0..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
native Declaration	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 A.80: BaseTypeDirectDefinition

Class	**BinaryManifestAddressableObject** (abstract)			
Note	This meta-class acts as an abstract base class for addressable objects in the context of the binary manifest of a CP software cluster. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BinaryManifestItem, BinaryManifestMetaDataField			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
address	Address	0..1	attr	This attribute specifies the address of the enclosing addressable object.
symbol	SymbolString	0..1	attr	This attribute specifies the symbol of the addressable object.

Table A.81: BinaryManifestAddressableObject

Class	**BinaryManifestItem**			
Note	This meta-class represents the ability to describe a specific handle or auxiliary field in the context of binary manifest resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItem.auxiliaryField, *BinaryManifestResource*.item			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
auxiliaryField	BinaryManifestItem	*	aggr	This aggregation is used to define structured Binary ManifestItems. **Tags:** atp.Splitkey=auxiliaryField, auxiliaryField.shortName xml.sequenceOffset=20
defaultValue	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a default value for a binary manifest handle or an auxiliaryField. This value shall be taken if no connection for this resource is possible. **Tags:** atp.Splitkey=defaultValue xml.sequenceOffset=10
isUnused	Boolean	0..1	attr	If true, the handle or auxiliary field in the context of binary manifest resource relates to an optional BinaryManifest ItemDefinition and is not used.
value	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a value for a binary manifest handle or an auxiliaryField. This value shall be taken to establish a connection. **Tags:** atp.Splitkey=value

Table A.82: BinaryManifestItem

Class	**BinaryManifestItemDefinition**			
Note	This meta-class provides the ability to define the handle definition or an auxiliary field of a binary manifest resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItemDefinition.auxiliaryFieldDefinition, BinaryManifestResourceDefinition.itemDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	BinaryManifestItemDefinition			
auxiliaryField Definition	BinaryManifestItem Definition	*	aggr	This aggregation is used to define structured Binary ManifestItemDefinitions. **Tags:** atp.Splitkey=auxiliaryFieldDefinition, auxiliaryField Definition.shortName
isOptional	Boolean	0..1	attr	If true, the handle definition or auxiliary field of a binary manifest resource is optional and may not be used in all BinaryManifestResources referring to this BinaryManifest ResourceDefinition.
size	PositiveInteger	0..1	attr	This attribute provides the ability to specify the size of the enclosing BinaryManifestResourceDefinition.

Table A.83: BinaryManifestItemDefinition

Class	BinaryManifestItemNumericalValue			
Note	This meta-class has the ability to provide a numerical value for a binary manifest item. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	Type	Mult.	Kind	Note
value	Numerical	0..1	attr	This attribute specifies the actual numerical value to be used in the binary manifest handle.

Table A.84: BinaryManifestItemNumericalValue

Class	BinaryManifestItemPointerValue			
Note	This meta-class has the ability to provide a value for a pointer in the context of a binary manifest item. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	Type	Mult.	Kind	Note
address	Address	0..1	attr	This attribute represents the address value of the enclosing pointer value.
symbol	SymbolString	0..1	attr	This attribute represents the symbol associated with the binary manifest handle.

Table A.85: BinaryManifestItemPointerValue

Class	BinaryManifestMetaDataField			
Note	This meta-class provides the ability to define a meta-data field for the binary manifest descriptor. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.metaDataField			
Attribute	Type	Mult.	Kind	Note
size	PositiveInteger	0..1	attr	The value of this attribute represents the size of the meta-data field in bytes.
value	VerbatimString	0..1	attr	This attribute specifies the value of the meta-data field.

Table A.86: BinaryManifestMetaDataField

Class	**BinaryManifestProvideResource**			
Note	This meta-class represents a provided resource in the binary manifest. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.provideResource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf NotifierSets	PositiveInteger	0..1	attr	This attribute provides an upper limit for the number of notifiers for this resource.
supports MultipleNotifier Sets	Boolean	0..1	attr	This attribute indicates whether the enclosing Binary ManifestResource supports multiple notifiers sets.

Table A.87: BinaryManifestProvideResource

Class	**BinaryManifestRequireResource**			
Note	This meta-class represents a required resource in the binary manifest. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.requireResource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connectionIs Mandatory	Boolean	0..1	attr	This attribute indicates whether the connection of the enclosing BinaryManifestResource is mandatory.

Table A.88: BinaryManifestRequireResource

Class	***BinaryManifestResource*** (abstract)			
Note	This meta-class acts as an abstract base class for specializations. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BinaryManifestProvideResource, BinaryManifestRequireResource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
globalResource Id	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines.
item (ordered)	BinaryManifestItem	*	aggr	This aggregation represents the collection of binary manifest handles owned by the enclosing binary manifest resource.
resource	CpSoftwareCluster Resource	0..1	ref	This reference identifies the CpSoftwareClusterResource (on design level) that corresponds to the BinaryManifest Resource (on integration level). **Tags:** atp.Splitkey=resource
resource Definition	BinaryManifest ResourceDefinition	0..1	ref	this reference identifies the definition of the Binary ManifestResource. The definition provides configuration information that is shared among all BinaryManifest Resources that refer to the BinaryManifestResource Definition. **Tags:** atp.Splitkey=resourceDefinition
resourceGuard Value	String	0..1	attr	This attribute specifies the guard value of the enclosing binary manifest resource.

Table A.89: BinaryManifestResource

Class	**BinaryManifestResourceDefinition**			
Note	This meta-class represents the ability to specify a resource definition that provides information that can be shared by all resources that refer to the respective resource definition. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.resourceDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
itemDefinition (ordered)	BinaryManifestItem Definition	*	aggr	This aggregation specifies the collection of handle definitions in the context of the enclosing binary manifest resource definitions.

Table A.90: BinaryManifestResourceDefinition

Class	**BlueprintMapping**			
Note	This meta-class represents the ability to map two an object and its blueprint.			
Base	*ARObject*, *AtpBlueprintMapping*			
Aggregated by	BlueprintMappingSet.blueprintMap			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blueprint	AtpBlueprint	0..1	ref	This represents the mapped blueprint. **Stereotypes:** atpIdentityContributor
derivedObject	Identifiable	0..1	ref	This represents the object which was derived from the blueprint. **Stereotypes:** atpIdentityContributor

Table A.91: BlueprintMapping

Class	***BlueprintPolicy*** (abstract)			
Note	This meta-class represents the ability to indicate whether blueprintable elements will be modifiable or not modifiable.			
Base	*ARObject*			
Subclasses	*BlueprintPolicyModifiable*, BlueprintPolicyNotModifiable			
Aggregated by	*AtpBlueprint*.blueprintPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
attributeName	String	0..1	attr	This identifies the related attribute of a BlueprintPolicy. For navigation over the model a subset of xpath expressions is used. **Stereotypes:** atpIdentityContributor

Table A.92: BlueprintPolicy

Class	**BlueprintPolicyNotModifiable**			
Note	The class represents that the related attribute is not modifiable during the blueprinting.			
Base	*ARObject*, *BlueprintPolicy*			
Aggregated by	*AtpBlueprint*.blueprintPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.93: BlueprintPolicyNotModifiable

Class	**BswAsynchronousServerCallPoint**			
Note	Represents an asynchronous procedure call point via the BSW Scheduler.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calledEntry	BswModuleClientServerEntry	0..1	ref	The entry to be called. **Tags:** atp.Splitkey=calledEntry

Table A.94: BswAsynchronousServerCallPoint

Class	**BswAsynchronousServerCallResultPoint**			
Note	The callback point for an BswAsynchronousServerCallPoint i.e. the point at which the result can be retrieved from the BSW Scheduler. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
asynchronous ServerCallPoint	BswAsynchronous ServerCallPoint	0..1	ref	The call point invoking the call to which the result belongs. **Tags:** atp.Splitkey=asynchronousServerCallPoint

Table A.95: BswAsynchronousServerCallResultPoint

Class	**BswAsynchronousServerCallReturnsEvent**			
Note	This is the "callback" event for asynchronous Client-Server-Communication via the BSW Scheduler which is thrown after completion of the asynchronous Client-Server call. Its eventSource specifies the call point to be used for retrieving the result.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventSource	BswAsynchronous ServerCallResultPoint	0..1	ref	The call point to be used for retrieving the result. **Tags:** atp.Splitkey=eventSource This Attribute is only used by the AUTOSAR Classic Platform.

Table A.96: BswAsynchronousServerCallReturnsEvent

Enumeration	**BswCallType**
Note	Denotes the mechanism by which the entry into the Bsw module shall be called.
Aggregated by	BswModuleEntry.callType
Literal	***Description***
callback	Callback (i.e. the caller specifies the signature) **Tags:** atp.EnumerationLiteralIndex=0
callout	Callout - provide defined means to extend the functionality of an existing module. In this case caller specifies the signature. **Tags:** atp.EnumerationLiteralIndex=4
interrupt	Interrupt routine **Tags:** atp.EnumerationLiteralIndex=1

▽

△

Enumeration	**BswCallType**
regular	Regular API call **Tags:** atp.EnumerationLiteralIndex=2
scheduled	Called by the scheduler **Tags:** atp.EnumerationLiteralIndex=3

Table A.97: BswCallType

Class	**BswCalledEntity**			
Note	BSW module entity which is designed to be called from another BSW module or cluster.			
Base	*ARObject*, *BswModuleEntity*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.entity			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.98: BswCalledEntity

Class	**BswCompositionTiming**			
Note	A model element used to define timing descriptions and constraints for a set of BswImplementations representing a BSW composition. A constraint defined at this level holds true for all referenced Bsw Implementations. Note, that multiple implementations of the same basic software module could be involved. TimingDescriptions aggregated by BswCompositionTiming are restricted to event chains referring to events which are derived from the class TDEventBswInternalBehavior and TDEventBsw. **Tags:** atp.recommendedPackage=TimingExtensions This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
implementation	BswImplementation	*	ref	This defines the scope of a BswCompositionTiming. All corresponding timing descriptions and constraints shall be defined within this scope. **Tags:** atp.Splitkey=implementation

Table A.99: BswCompositionTiming

Class	**BswDataReceivedEvent**			
Note	This event is thrown on reception of the referenced data via Sender-Receiver-Communication over the BSW Scheduler.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
data	VariableDataPrototype	0..1	ref	The received data. **Tags:** atp.Splitkey=data

Table A.100: BswDataReceivedEvent

Class	***BswDataReceptionPolicy*** (abstract)			
Note	Specifies the reception policy for the referred data in sender-receiver communication over the BSW Scheduler. To be used for inter-partition and/or inter-core communication.			
Base	*ARObject*, *BswApiOptions*			
Subclasses	BswQueuedDataReceptionPolicy			
Aggregated by	BswInternalBehavior.receptionPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
receivedData	VariableDataPrototype	0..1	ref	The data received over the BSW Scheduler using this policy. **Tags:** atp.Splitkey=receivedData

Table A.101: BswDataReceptionPolicy

Class	**BswDirectCallPoint**			
Note	Represents a concrete point in the code from where a BswModuleEntry is called directly, i.e. not via the BSW Scheduler. This information can be used to analyze call tree and resource locking scenarios. It is not needed to configure the BSW Scheduler.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calledEntry	BswModuleEntry	0..1	ref	The BswModuleEntry called at this point. **Tags:** atp.Splitkey=calledEntry This Attribute is only used by the AUTOSAR Classic Platform.
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order. **Tags:** atp.Splitkey=calledFromWithinExclusiveArea

Table A.102: BswDirectCallPoint

Class	**BswDistinguishedPartition**			
Note	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.			
Base	*ARObject*, *Referrable*			
Aggregated by	BswInternalBehavior.distinguishedPartition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.103: BswDistinguishedPartition

Class	**BswEntryRelationship**			
Note	Describes a relationship between two BswModuleEntrys and the type of relationship.			
Base	*ARObject*			
Aggregated by	BswEntryRelationshipSet.bswEntryRelationship			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**BswEntryRelationship**			
bswEntry Relationship Type	BswEntryRelationship Enum	0..1	attr	Denotes the type of the relationship. **Tags:** xml.sequenceOffset=5
from	BswModuleEntry	0..1	ref	Type of relationship that refers to the abstract BswModule Entry. Please notice that in this case the bswEntry RelationshipType shall be set to drivedFrom. **Tags:** atp.Splitkey=from This Attribute is only used by the AUTOSAR Classic Platform.
to	BswModuleEntry	0..1	ref	Type of relationship that refers to the concrete Bsw ModuleEntry **Tags:** atp.Splitkey=to This Attribute is only used by the AUTOSAR Classic Platform.

Table A.104: BswEntryRelationship

Class	**BswEntryRelationshipSet**			
Note	Describes a set of relationships between two BswModuleEntrys. **Tags:** atp.recommendedPackage=BswEntryRelationshipSets This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswEntry Relationship	BswEntryRelationship	*	aggr	Relationship between two BswModuleEntrys. **Tags:** atp.Splitkey=bswEntryRelationship

Table A.105: BswEntryRelationshipSet

Class	***BswEvent*** (abstract)			
Note	Base class of various kinds of events which are used to trigger a BswModuleEntity of this BSW module or cluster. The event is local to the BSW module or cluster. The short name of the meta-class instance is intended as an input to configure the required API of the BSW Scheduler.			
Base	*ARObject*, *AbstractEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswInterruptEvent, BswOperationInvokedEvent, *BswScheduleEvent*			
Aggregated by	BswInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the usage of the event is limited to the context of the referred Bsw DistinguishedPartitions. **Tags:** atp.Splitkey=contextLimitation
disabledInMode	ModeDeclaration	*	iref	The modes, in which this event is disabled. **Tags:** atp.Splitkey=disabledInMode, disabledIn Mode.contextModeDeclarationGroup, disabledIn Mode.targetMode **InstanceRef implemented by:** ModeInBswModule DescriptionInstanceRef
startsOnEvent	BswModuleEntity	0..1	ref	The entity which is started by the event. **Tags:** atp.Splitkey=startsOnEvent

Table A.106: BswEvent

Class	**BswExclusiveAreaPolicy**			
Note	The ExclusiveArea for which the BSW Scheduler using this policy.			
Base	*ARObject*, *BswApiOptions*			
Aggregated by	BswInternalBehavior.exclusiveAreaPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
apiPrinciple	ApiPrincipleEnum	0..1	attr	Specifies for this ExclusiveArea if either one common set of Enter and Exit APIs for the whole BSW module is requested from the SchM or if the set of Enter and Exit APIs is expected per BswModuleEntity. The default value is "common".
exclusiveArea	ExclusiveArea	0..1	ref	The ExclusiveArea for which the BSW Scheduler using this policy. **Tags:** atp.Splitkey=exclusiveArea

Table A.107: BswExclusiveAreaPolicy

Enumeration	**BswExecutionContext**
Note	Specifies the execution context required or guaranteed for the call associated with this service.
Aggregated by	BswModuleEntry.executionContext
Literal	***Description***
hook	Context of an OS "hook" routine always **Tags:** atp.EnumerationLiteralIndex=0
interruptCat1	CAT1 interrupt context always **Tags:** atp.EnumerationLiteralIndex=1
interruptCat2	CAT2 interrupt context always **Tags:** atp.EnumerationLiteralIndex=2
task	Task context always **Tags:** atp.EnumerationLiteralIndex=3
unspecified	The execution context is not specified by the API **Tags:** atp.EnumerationLiteralIndex=4

Table A.108: BswExecutionContext

Class	**BswExternalTriggerOccurredEvent**			
Note	A BswEvent resulting from a trigger released by another module or cluster.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
trigger	Trigger	0..1	ref	The trigger associated with this event. The trigger is external to this module. **Tags:** atp.Splitkey=trigger

Table A.109: BswExternalTriggerOccurredEvent

Class	**BswImplementation**			
Note	Contains the implementation specific information in addition to the generic specification (BswModule Description and BswBehavior). It is possible to have several different BswImplementations referring to the same BswBehavior. **Tags:** atp.recommendedPackage=BswImplementations This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *Implementation*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arRelease Version	RevisionLabelString	0..1	attr	Version of the AUTOSAR Release on which this implementation is based. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.
behavior	BswInternalBehavior	0..1	ref	The behavior of this implementation. This relation is made as an association because • it follows the pattern of the SWCT • since ARElement cannot be split, but we want supply the implementation later, the BswImplementation is not aggregated in BswBehavior **Tags:** atp.Splitkey=behavior
preconfigured Configuration	EcucModule ConfigurationValues	*	ref	Reference to the set of preconfigured (i.e. fixed) configuration values for this BswImplementation. If the BswImplementation represents a cluster of several modules, more than one EcucModuleConfigurationValues element can be referred (at most one per module), otherwise at most one such element can be referred. **Tags:** atp.Splitkey=preconfiguredConfiguration xml.roleWrapperElement=true
recommended Configuration	EcucModule ConfigurationValues	*	ref	Reference to one or more sets of recommended configuration values for this module or module cluster. **Tags:** atp.Splitkey=recommendedConfiguration
vendorApiInfix	Identifier	0..1	attr	In driver modules which can be instantiated several times on a single ECU, SRS_BSW_00347 requires that the names of files, APIs, published parameters and memory allocation keywords are extended by the vendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific API name is generated as follows: <Module Name>_<vendorId>_ <vendorApiInfix>_<API name from SWS>. E.g. assuming that the vendorId of the implementer is 123 and the implementer chose a vendorApiInfix of "v11r456" an API name Can_Write defined in the SWS will translate to Can_123_v11r456_Write. This attribute is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1. See also SWS_BSW_00102.

▽

△

Class	**BswImplementation**			
vendorSpecific ModuleDef	EcucModuleDef	*	ref	Reference to • the vendor specific EcucModuleDef used in this Bsw Implementation if it represents a single module • several EcucModuleDefs used in this Bsw Implementation if it represents a cluster of modules • one or no EcucModuleDefs used in this Bsw Implementation if it represents a library **Tags:** atp.Splitkey=vendorSpecificModuleDef xml.roleWrapperElement=true

Table A.110: BswImplementation

Class	**BswInternalBehavior**			
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*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.internalBehavior			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory, arTypedPer InstanceMemory.shortName, arTypedPerInstance Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
bswPerInstance MemoryPolicy	BswPerInstance MemoryPolicy	*	aggr	Policy for a arTypedPerInstanceMemory The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswPerInstanceMemoryPolicy, bswPer InstanceMemoryPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
clientPolicy	BswClientPolicy	*	aggr	Policy for a requiredClientServerEntry. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clientPolicy, clientPolicy.variationPoint.short Label vh.latestBindingTime=preCompileTime

▽

△

Class	**BswInternalBehavior**			
distinguished Partition	BswDistinguished Partition	*	aggr	Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=distinguishedPartition, distinguished Partition.shortName, distinguishedPartition.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
entity	BswModuleEntity	*	aggr	A code entity for which the behavior is described **Stereotypes:** atpVariation **Tags:** atp.Splitkey=entity, entity.shortName, entity.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=5
event	BswEvent	*	aggr	An event required by this module behavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=event, event.shortName, event.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
exclusiveArea Policy	BswExclusiveArea Policy	*	aggr	Policy for an ExclusiveArea in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a basic software module for its implementation. **Tags:** atp.Splitkey=includedDataTypeSet
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Tags:** atp.Splitkey=includedModeDeclarationGroupSet
internal TriggeringPoint	BswInternalTriggering Point	*	aggr	An internal triggering point. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint, internalTriggering Point.shortName, internalTriggeringPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=2 This Attribute is only used by the AUTOSAR Classic Platform.
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:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPointPolicy, internal TriggeringPointPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**BswInternalBehavior**			
modeReceiver Policy	BswModeReceiver Policy	*	aggr	Implementation policy for the reception of mode switches. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeReceiverPolicy, modeReceiver Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
modeSender Policy	BswModeSenderPolicy	*	aggr	Implementation policy for providing a mode group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeSenderPolicy, modeSender Policy.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:** atpVariation **Tags:** atp.Splitkey=parameterPolicy, parameterPolicy.variation Point.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:** atpVariation **Tags:** atp.Splitkey=perInstanceParameter, perInstance Parameter.shortName, perInstanceParameter.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
receptionPolicy	BswDataReception Policy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=receptionPolicy, receptionPolicy.variation Point.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:** atpVariation **Tags:** atp.Splitkey=releasedTriggerPolicy, releasedTrigger Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**BswInternalBehavior**			
schedulerName Prefix	BswSchedulerName Prefix	*	aggr	Optional definition of one or more prefixes to be used for the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=schedulerNamePrefix, schedulerName Prefix.shortName, schedulerNamePrefix.variation Point.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:** atpVariation **Tags:** atp.Splitkey=sendPolicy, sendPolicy.variationPoint.short Label 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:** atpVariation **Tags:** atp.Splitkey=serviceDependency, service Dependency.ident.shortName, service Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
triggerDirect Implementation	BswTriggerDirect Implementation	*	aggr	Specifies a trigger to be directly implemented via OS calls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=triggerDirectImplementation, triggerDirect Implementation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=15
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation. **Tags:** atp.Splitkey=variationPointProxy, variationPoint Proxy.shortName

Table A.111: BswInternalBehavior

Class	**BswInternalTriggerOccurredEvent**			
Note	A BswEvent, which can happen sporadically. The event is activated by explicit calls from the module to the BSW Scheduler. The main purpose for such an event is to cause a context switch, e.g. from an ISR context into a task context. Activation and switching are handled within the same module or cluster only.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**BswInternalTriggerOccurredEvent**			
eventSource	BswInternalTriggering Point	0..1	ref	The activation point is the source of this event. **Tags:** atp.Splitkey=eventSource This Attribute is only used by the AUTOSAR Classic Platform.

Table A.112: BswInternalTriggerOccurredEvent

Class	**BswInternalTriggeringPoint**			
Note	Represents the activation point for one or more BswInternalTriggerOccurredEvents. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.internalTriggeringPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, specifies a queued processing of the internal trigger event.

Table A.113: BswInternalTriggeringPoint

Enumeration	**BswInterruptCategory**
Note	Category of the interrupt service
Aggregated by	BswInterruptEntity.interruptCategory
Literal	***Description***
cat1	Cat1 interrupt routines are not controlled by the OS and are only allowed to make a very limited selection of OS calls to enable and disable all interrupts. The BswInterruptEntity is implemented by the interrupt service routine, which is directly called from the interrupt vector (not via the OS). **Tags:** atp.EnumerationLiteralIndex=0
cat2	Cat2 interrupt routines are controlled by the OS and they are allowed to make OS calls. The Bsw InterruptEntity is implemented by the interrupt handler, which is called from the OS. **Tags:** atp.EnumerationLiteralIndex=1

Table A.114: BswInterruptCategory

Class	**BswInterruptEntity**			
Note	BSW module entity, which is designed to be triggered by an interrupt.			
Base	*ARObject*, *BswModuleEntity*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.entity			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interrupt Category	BswInterruptCategory	0..1	attr	Category of the interrupt
interruptSource	String	0..1	attr	Allows a textual documentation of the intended interrupt source.

Table A.115: BswInterruptEntity

Class	BswInterruptEvent			
Note	This meta-class represents an event triggered by an interrupt.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.116: BswInterruptEvent

Class	BswMgrNeeds			
Note	Specifies the abstract needs on the configuration of the Basic Software Manager for one "user".			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.117: BswMgrNeeds

Class	BswModeManagerErrorEvent			
Note	This represents the ability to react on errors occurring during mode handling.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	This represents the ModeDeclarationGroupPrototype for which the error behavior of the mode manager applies. **Tags:** atp.Splitkey=modeGroup

Table A.118: BswModeManagerErrorEvent

Class	BswModeReceiverPolicy			
Note	Specifies the details for the reception of a mode switch for the referred mode group.			
Base	*ARObject*			
Aggregated by	BswInternalBehavior.modeReceiverPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to TRUE the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
requiredMode Group	ModeDeclarationGroup Prototype	0..1	ref	The required mode group for which the policy is specified. **Tags:** atp.Splitkey=requiredModeGroup
supports Asynchronous ModeSwitch	Boolean	0..1	attr	Specifies whether the module can handle the reception of an asynchronous mode switch (true) or not (false).

Table A.119: BswModeReceiverPolicy

Class	**BswModeSenderPolicy**			
Note	Specifies the details for the sending of a mode switch for the referred mode group.			
Base	*ARObject*			
Aggregated by	BswInternalBehavior.modeSenderPolicy			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ackRequest	BswModeSwitchAck Request	0..1	aggr	Request for acknowledgement **Tags:** atp.Splitkey=ackRequest
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to TRUE the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
providedMode Group	ModeDeclarationGroup Prototype	0..1	ref	The provided mode group for which the policy is specified. **Tags:** atp.Splitkey=providedModeGroup
queueLength	PositiveInteger	0..1	attr	Length of call queue on the sender side. The queue is implemented by the RTE resp.BswScheduler. The value shall be greater or equal to 0. Setting the value of queue Length to 0 implies non-queued communication.

Table A.120: BswModeSenderPolicy

Class	**BswModeSwitchAckRequest**			
Note	Requests acknowledgements that a mode switch has been processed successfully			
Base	*ARObject*			
Aggregated by	BswModeSenderPolicy.ackRequest			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported.

Table A.121: BswModeSwitchAckRequest

Class	**BswModeSwitchEvent**			
Note	A BswEvent resulting from a mode switch.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
activation	ModeActivationKind	0..1	attr	Kind of activation w.r.t. to the referred mode.
mode (ordered)	ModeDeclaration	0..2	iref	Reference to one or two Modes that initiate the Mode Switch Event. **InstanceRef implemented by:** ModeInBswModule DescriptionInstanceRef

Table A.122: BswModeSwitchEvent

Class	**BswModeSwitchedAckEvent**			
Note	The event is raised after a switch of the referenced mode group has been acknowledged or an error occurs. The referenced mode group shall be provided by this module.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**BswModeSwitchedAckEvent**			
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	A mode group provided by this module. The acknowledgement of a switch of this group raises this event. **Tags:** atp.Splitkey=modeGroup

Table A.123: BswModeSwitchedAckEvent

Class	***BswModuleCallPoint*** (abstract)			
Note	Represents a point at which a BswModuleEntity handles a procedure call into a BswModuleEntry, either directly or via the BSW Scheduler.			
Base	*ARObject*, *Referrable*			
Subclasses	BswAsynchronousServerCallPoint, BswAsynchronousServerCallResultPoint, BswDirectCallPoint, Bsw SynchronousServerCallPoint			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the call point is used only in the context of the referred Bsw DistinguishedPartitions. **Tags:** atp.Splitkey=contextLimitation

Table A.124: BswModuleCallPoint

Class	**BswModuleClientServerEntry**			
Note	This meta-class represents a single API entry into the BSW module or cluster that has the ability to be called in client-server fashion via the BSW Scheduler. In this regard it is more special than BswModuleEntry and can be seen as a wrapper around the Bsw ModuleEntry to which it refers (property encapsulatedEntry). **Tags:** atp.recommendedPackage=BswModuleEntrys			
Base	*ARObject*, *Referrable*			
Aggregated by	BswModuleDescription.providedClientServerEntry, BswModuleDescription.requiredClientServerEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
encapsulated Entry	BswModuleEntry	0..1	ref	The underlying BswModuleEntry. **Tags:** atp.Splitkey=encapsulatedEntry xml.sequenceOffset=5 This Attribute is only used by the AUTOSAR Classic Platform.
isReentrant	Boolean	0..1	attr	Reentrancy from the viewpoint of clients invoking the service via the BSW Scheduler: • true: Enables the service to be invoked again, before the service has finished. • false: It is prohibited to invoke the service again before is has finished. **Tags:** xml.sequenceOffset=10

▽

△

Class	**BswModuleClientServerEntry**			
isSynchronous	Boolean	0..1	attr	Synchronicity from the viewpoint of clients invoking the service via the BSW Scheduler: • true: This calls a synchronous service, i.e. the service is completed when the call returns. • false: The service (on semantical level) may not be complete when the call returns. **Tags:** xml.sequenceOffset=15

Table A.125: BswModuleClientServerEntry

Class	**BswModuleDependency**			
Note	This class collects the dependencies of a BSW module or cluster on a certain other BSW module. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswModuleDescription.bswModuleDependency			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
targetModuleId	PositiveInteger	0..1	attr	AUTOSAR identifier of the target module of which the dependencies are defined. This information is optional, because the target module may also be identified by targetModuleRef. **Tags:** xml.sequenceOffset=5
targetModule Ref	BswModuleDescription	0..1	ref	Reference to the target module. It is an <<atpUriDef>> because the reference shall be used to identify the target module without actually needing the description of that target module. **Stereotypes:** atpUriDef; atpVariation **Tags:** atp.Splitkey=targetModuleRef, targetModuleRef.bsw ModuleDescription, targetModuleRef.variationPoint.short Label vh.latestBindingTime=preCompileTime xml.sequenceOffset=7

Table A.126: BswModuleDependency

Class	**BswModuleDescription**			
Note	Root element for the description of a single BSW module or BSW cluster. In case it describes a BSW module, the short name of this element equals the name of the BSW module. **Tags:** atp.recommendedPackage=BswModuleDescriptions This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**BswModuleDescription**			
bswModule Dependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswModuleDependency, bswModule Dependency.shortName, bswModule Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
bswModule Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the BSW module. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswModuleDocumentation, bswModule Documentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6
expectedEntry	BswModuleEntry	*	ref	Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=expectedEntry, expectedEntry.bswModule Entry, expectedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	*	ref	Specifies an entry provided by this module which can be called by other modules. This includes "main" functions, interrupt routines, and callbacks. Replacement of providedEntry / expectedCallback. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=implementedEntry, implementedEntry.bsw ModuleEntry, implementedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internalBehavior	BswInternalBehavior	*	aggr	The various BswInternalBehaviors associated with a Bsw ModuleDescription can be distributed over several physical files. Therefore the aggregation is <<atp Splitable>>. **Tags:** atp.Splitkey=internalBehavior, internalBehavior.short Name xml.sequenceOffset=65
moduleId	PositiveInteger	0..1	attr	Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen. **Tags:** xml.sequenceOffset=5
providedClient ServerEntry	BswModuleClientServer Entry	*	aggr	Specifies that this module provides a client server entry which can be called from another partition or core.This entry is declared locally to this context and will be connected to the requiredClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedClientServerEntry, providedClient ServerEntry.shortName, providedClientServer Entry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45

▽

△

Class	**BswModuleDescription**			
providedData	VariableDataPrototype	*	aggr	Specifies a data prototype provided by this module in order to be read from another partition or core.The providedData is declared locally to this context and will be connected to the requiredData of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedData, providedData.shortName, providedData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
providedMode Group	ModeDeclarationGroup Prototype	*	aggr	A set of modes which is owned and provided by this module or cluster. It can be connected to the required ModeGroups of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with modes provided via ports by an associated ServiceSwComponentType, EcuAbstraction SwComponentType or ComplexDeviceDriverSw ComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedModeGroup, providedMode Group.shortName, providedModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
releasedTrigger	Trigger	*	aggr	A Trigger released by this module or cluster. It can be connected to the requiredTriggers of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with Triggers provided via ports by an associated ServiceSwComponentType, Ecu AbstractionSwComponentType or ComplexDeviceDriver SwComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=releasedTrigger, releasedTrigger.shortName, releasedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35
requiredClient ServerEntry	BswModuleClientServer Entry	*	aggr	Specifies that this module requires a client server entry which can be implemented on another partition or core.This entry is declared locally to this context and will be connected to the providedClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredClientServerEntry, requiredClient ServerEntry.shortName, requiredClientServer Entry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

▽

△

Class	**BswModuleDescription**			
requiredData	VariableDataPrototype	*	aggr	Specifies a data prototype required by this module in oder to be provided from another partition or core.The required Data is declared locally to this context and will be connected to the providedData of another or the same module via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredData, requiredData.shortName, requiredData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
requiredMode Group	ModeDeclarationGroup Prototype	*	aggr	Specifies that this module or cluster depends on a certain mode group. The requiredModeGroup is local to this context and will be connected to the providedModeGroup of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredModeGroup, requiredMode Group.shortName, requiredModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30
requiredTrigger	Trigger	*	aggr	Specifies that this module or cluster reacts upon an external trigger.This requiredTrigger is declared locally to this context and will be connected to the providedTrigger of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredTrigger, requiredTrigger.shortName, requiredTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40

Table A.127: BswModuleDescription

Class	***BswModuleEntity*** (abstract)			
Note	Specifies the smallest code fragment which can be described for a BSW module or cluster within AUTOSAR.			
Base	*ARObject*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswCalledEntity, BswInterruptEntity, BswSchedulableEntity			
Aggregated by	BswInternalBehavior.entity			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is accessed via API call by this entity. It shall be a ModeDeclarationGroupPrototype required by this module or cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=accessedModeGroup, accessedMode Group.modeDeclarationGroupPrototype, accessedMode Group.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	***BswModuleEntity*** (abstract)			
activationPoint	BswInternalTriggering Point	*	ref	Activation point used by the module entity to activate one or more internal triggers. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=activationPoint, activationPoint.bswInternal TriggeringPoint, activationPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
callPoint	BswModuleCallPoint	*	aggr	A call point used in the code of this entity. The variability of this association is especially targeted at debug scenarios: It is possible to have one variant calling into the AUTOSAR debug module and another one which doesn't. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=callPoint, callPoint.shortName, call Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive Point	BswVariableAccess	*	aggr	The data is received via the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReceivePoint, dataReceivePoint.short Name, dataReceivePoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataSendPoint	BswVariableAccess	*	aggr	The data is sent via the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataSendPoint, dataSendPoint.shortName, dataSendPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	0..1	ref	The entry which is implemented by this module entity. **Tags:** atp.Splitkey=implementedEntry This Attribute is only used by the AUTOSAR Classic Platform.
issuedTrigger	Trigger	*	ref	A trigger issued by this entity via BSW Scheduler API call. It shall be a BswTrigger released (i.e. owned) by this module or cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=issuedTrigger, issuedTrigger.trigger, issued Trigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
managedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is managed by this entity. It shall be a ModeDeclarationGroupPrototype provided by this module or cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=managedModeGroup, managedMode Group.modeDeclarationGroupPrototype, managedMode Group.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	***BswModuleEntity*** (abstract)			
schedulerName Prefix	BswSchedulerName Prefix	0..1	ref	A prefix to be used in generated names for the Bsw ModuleScheduler in the context of this BswModuleEntity, for example entry point prototypes, macros for dealing with exclusive areas, header file names. Details are defined in the SWS RTE. The prefix supersedes default rules for the prefix of those names. **Tags:** atp.Splitkey=schedulerNamePrefix

Table A.128: BswModuleEntity

Class	**BswModuleEntry**			
Note	This class represents a single API entry (C-function prototype) into the BSW module or cluster. The name of the C-function is equal to the short name of this element with one exception: In case of multiple instances of a module on the same CPU, special rules for "infixes" apply, see description of class BswImplementation. **Tags:** atp.recommendedPackage=BswModuleEntrys This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
argument (ordered)	SwServiceArg	*	aggr	An argument belonging to this BswModuleEntry. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=45
bswEntryKind	BswEntryKindEnum	0..1	attr	This describes whether the entry is concrete or abstract. If the attribute is missing the entry is considered as concrete. **Tags:** xml.sequenceOffset=40
callType	BswCallType	0..1	attr	The type of call associated with this service. **Tags:** xml.sequenceOffset=25
execution Context	BswExecutionContext	0..1	attr	Specifies the execution context which is required (in case of entries into this module) or guaranteed (in case of entries called from this module) for this service. **Tags:** xml.sequenceOffset=30
function Prototype Emitter	NameToken	0..1	attr	This attribute is used to control the generation of function prototypes. If set to "RTE", the RTE generates the function prototypes in the Module Interlink Header File.
isReentrant	Boolean	0..1	attr	Reentrancy from the viewpoint of function callers: • true: Enables the service to be invoked again, before the service has finished. • false: It is prohibited to invoke the service again before is has finished. **Tags:** xml.sequenceOffset=15

▽

△

Class	**BswModuleEntry**			
isSynchronous	Boolean	0..1	attr	Synchronicity from the viewpoint of function callers: • true: This calls a synchronous service, i.e. the service is completed when the call returns. • false: The service (on semantical level) may not be complete when the call returns. **Tags:** xml.sequenceOffset=20
returnType	SwServiceArg	0..1	aggr	The return type belonging to this bswModuleEntry. **Tags:** atp.Splitkey=returnType, returnType.shortName xml.sequenceOffset=40
role	Identifier	0..1	attr	Specifies the role of the entry in the given context. It shall be equal to the standardized name of the service call, especially in cases where no ServiceIdentifier is specified, e.g. for callbacks. Note that the ShortName is not always sufficient because it maybe vendor specific (e.g. for callbacks which can have more than one instance). **Tags:** xml.sequenceOffset=10
serviceId	PositiveInteger	0..1	attr	Refers to the service identifier of the Standardized Interfaces of AUTOSAR basic software. For non-standardized interfaces, it can optionally be used for proprietary identification. **Tags:** xml.sequenceOffset=5
swServiceImpl Policy	SwServiceImplPolicy Enum	0..1	attr	Denotes the implementation policy as a standard function call, inline function or macro. This has to be specified on interface level because it determines the signature of the call. **Tags:** xml.sequenceOffset=35

Table A.129: BswModuleEntry

Class	**BswModuleTiming**			
Note	A model element used to define timing descriptions and constraints for the BswInternalBehavior of one BSW Module. Thereby, for each BswInternalBehavior a separate timing can be specified. A constraint defined at this level holds true for all Implementations of that BswInternalBehavior. TimingDescriptions aggregated by BswModuleTiming are restricted to event chains referring to events which are derived from the class TDEventBswInternalBehavior. **Tags:** atp.recommendedPackage=TimingExtensions This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
behavior	BswInternalBehavior	0..1	ref	This defines the scope of a BswModuleTiming. All corresponding timing descriptions and constraints shall be defined within this scope. **Tags:** atp.Splitkey=behavior

Table A.130: BswModuleTiming

Class	BswOperationInvokedEvent			
Note	This event is thrown on operation invocation in Client-Server-Communication via the BSW Scheduler. Its "entry" reference provides the BswClientServerEntry that is called subsequently. Note this event is not needed in case of direct function calls.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
entry	BswModuleClientServerEntry	0..1	ref	The providedClientServerEntry invoked by this event. **Tags:** atp.Splitkey=entry

Table A.131: BswOperationInvokedEvent

Class	BswQueuedDataReceptionPolicy			
Note	Reception policy attributes specific for queued receiving.			
Base	*ARObject*, *BswApiOptions*, *BswDataReceptionPolicy*			
Aggregated by	BswInternalBehavior.receptionPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
queueLength	PositiveInteger	0..1	attr	Length of queue for received events.

Table A.132: BswQueuedDataReceptionPolicy

Class	BswSchedulableEntity			
Note	BSW module entity, which is designed for control by the BSW Scheduler. It may for example implement a so-called "main" function.			
Base	*ARObject*, *BswModuleEntity*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.entity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.133: BswSchedulableEntity

Class	***BswScheduleEvent*** (abstract)			
Note	BswEvent that is able to start a BswSchedulabeEntity.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswAsynchronousServerCallReturnsEvent, BswBackgroundEvent, BswDataReceivedEvent, BswExternalTriggerOccurredEvent, BswInternalTriggerOccurredEvent, BswModeManagerErrorEvent, BswModeSwitchEvent, BswModeSwitchedAckEvent, BswOsTaskExecutionEvent, BswTimingEvent			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.134: BswScheduleEvent

Class	BswServiceDependency			
Note	Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.			
Base	*ARObject*, *ServiceDependency*			
Aggregated by	BswInternalBehavior.serviceDependency			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**BswServiceDependency**			
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object (owned by this module or cluster) in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime
assignedEntry Role	RoleBasedBswModule EntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedEntryRole, assignedEntry Role.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ident	BswService DependencyIdent	0..1	aggr	This adds the ability to become referrable to BswService Dependency. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=-100
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds. **Tags:** atp.Splitkey=serviceNeeds, serviceNeeds.short Name

Table A.135: BswServiceDependency

Class	**BswServiceDependencyIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable BswServiceDependency.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *IdentCaption*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswServiceDependency.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.136: BswServiceDependencyIdent

Class	**BswSynchronousServerCallPoint**			
Note	Represents a synchronous procedure call point via the BSW Scheduler.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calledEntry	BswModuleClientServer Entry	0..1	ref	The entry to be called. **Tags:** atp.Splitkey=calledEntry
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order. **Tags:** atp.Splitkey=calledFromWithinExclusiveArea

Table A.137: BswSynchronousServerCallPoint

Class	BswTimingEvent			
Note	A recurring BswEvent driven by a time period.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
period	TimeValue	0..1	attr	Requirement for the time period (in seconds) by which this event is triggered.

Table A.138: BswTimingEvent

Class	BswTriggerDirectImplementation			
Note	Specifies a released trigger to be directly implemented via OS calls, for example in a Complex Driver module.			
Base	*ARObject*			
Aggregated by	BswInternalBehavior.triggerDirectImplementation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cat2Isr	Identifier	0..1	attr	The name of the OS category 2 ISR, which is controlled by the referred trigger. This means, that the module manages the category 2 ISR (e.g. according hardware initialization and enabling of ISR). Instead of calling an RTE / SchM API to raise the appropriate events in components or modules receiving the trigger, this ISR directly schedules the triggered ExecutableEntitys. The ISR name is required by the integrator to map the Bsw Events and RTEEvents to this ISR.
masteredTrigger	Trigger	0..1	ref	The trigger which is directly mastered by this module. There may be several different BswTriggerDirect Implementations mastering the same Trigger. This may be required e.g. due to memory partitioning. **Tags:** atp.Splitkey=masteredTrigger
task	Identifier	0..1	attr	The name of the OS task, which is controlled by the referred trigger. This means, that the module uses the trigger condition to directly activate an OS task instead of calling an API of the BswScheduler. The task name is required by the RTE generator resp. BswScheduler to raise the appropriate events in components or modules receiving the trigger.

Table A.139: BswTriggerDirectImplementation

Class	BswVariableAccess			
Note	The presence of a BswVariableAccess implies that a BswModuleEntity needs access to a VariableData Prototype via the BSW Scheduler. The kind of access is specified by the role in which the class is used.			
Base	*ARObject*, *Referrable*			
Aggregated by	*BswModuleEntity*.dataReceivePoint, *BswModuleEntity*.dataSendPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessed Variable	VariableDataPrototype	0..1	ref	The data accessed via the BSW Scheduler. **Tags:** atp.Splitkey=accessedVariable

▽

△

Class	**BswVariableAccess**			
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the variable is received resp. sent only in the context of the referred BswDistinguishedPartitions. **Tags:** atp.Splitkey=contextLimitation

Table A.140: BswVariableAccess

Class	**BufferProperties**			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	*ARObject*			
Aggregated by	TransformationTechnology.bufferProperties			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
headerLength	Integer	0..1	attr	Defines the length of the header (in bits) this transformer will add in front of the data.
inPlace	Boolean	0..1	attr	If set, the transformer uses the input buffer as output buffer.

Table A.141: BufferProperties

Class	**BulkNvDataDescriptor**			
Note	This meta-class represents one bulk NV Data Block that is read-only for the application software. The purpose of a bulk NV Data Block is to provide access to information uploaded to the vehicle at e.g. the end of the production line.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, NvBlockSwComponentType.bulkNvDataDescriptor			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bulkNvBlock	VariableDataPrototype	0..1	aggr	This aggregation represents the actual bulk NVBlock. **Tags:** atp.Splitkey=bulkNvBlock, bulkNvBlock.shortName
nvBlockData Mapping	NvBlockDataMapping	*	aggr	Defines the mapping between the VariableData Prototypes in the NvBlockComponents ports and the VariableDataPrototypes of the non-volatile memory. The aggregation of NvBlockDataMapping is subject to variability with the purpose to support the conditional existence of nv data ports. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nvBlockDataMapping, nvBlockData Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

Table A.142: BulkNvDataDescriptor

Class	**BurstPatternEventTriggering**
Note	Describes the maximum number of occurrences of the same event in a given time interval. Typically used to model a worst case activation scenario.
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Timing Constraint*, *Traceable*
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement

▽

△

Class	**BurstPatternEventTriggering**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxNumberOf Occurrences	PositiveInteger	0..1	attr	The maximum number of event occurrences within the given time interval. The event may never occur, or may occur N times between 1 and `maxNumberOfOccurrences`. If the parameter `minNumberOfOccurrences` is specified then the event occurs at least the number of times specified by `minNumberOfOccurrences` and at maximum by `maxNumberOfOccurrences`.
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	Specifies the minimum distance between subsequent occurrences of the event within the given time interval. **Tags:** atp.Splitkey=minimumInterArrivalTime
minNumberOf Occurrences	PositiveInteger	0..1	attr	The minimum number of event occurrences within the given time interval. **Tags:** xml.sequenceOffset=10
patternJitter	MultidimensionalTime	0..1	aggr	The maximum deviation of the time interval's starting point from the beginning of the given period. This parameter is only applicable in conjunction with the parameter `patternPeriod` **Tags:** atp.Splitkey=patternJitter
patternLength	MultidimensionalTime	0..1	aggr	The duration of the time interval within which the event repeatedly occurs. The event occurs at arbitrary points in time within the given time interval. **Tags:** atp.Splitkey=patternLength
patternPeriod	MultidimensionalTime	0..1	aggr	The time distance between the beginnings of subsequent repetitions of the given burst pattern. **Tags:** atp.Splitkey=patternPeriod

Table A.143: BurstPatternEventTriggering

Class	**BusMirrorCanIdRangeMapping**			
Note	This element defines a rule for remapping a set of CAN IDs.			
Base	*ARObject*			
Aggregated by	BusMirrorChannelMappingCan.canIdRangeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationBase Id	PositiveInteger	0..1	attr	Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.
sourceCanId Code	PositiveInteger	0..1	attr	Value to match masked original CAN IDs.
sourceCanId Mask	PositiveInteger	0..1	attr	Mask applied to original CAN IDs before comparison.

Table A.144: BusMirrorCanIdRangeMapping

Class	**BusMirrorCanIdToCanIdMapping**			
Note	This element defines a rule for remapping a single CAN ID.			
Base	*ARObject*			
Aggregated by	BusMirrorChannelMappingCan.canIdToCanIdMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
remappedCanId	PositiveInteger	0..1	attr	This attribute defines the CanId on the targetChannel.

▽

△

Class	**BusMirrorCanIdToCanIdMapping**			
sourceCanId	CanFrameTriggering	0..1	ref	This reference points to the sourceFrame with sourceCanId on the sourceChannel. **Tags:** atp.Splitkey=sourceCanId

Table A.145: BusMirrorCanIdToCanIdMapping

Class	**BusMirrorChannel**			
Note	This element assigns a busMirrorNetworkId to the referenced channel.			
Base	*ARObject*			
Aggregated by	*BusMirrorChannelMapping*.sourceChannel, *BusMirrorChannelMapping*.targetChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
busMirror NetworkId	PositiveInteger	0..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:** atp.Splitkey=channel, channel.physicalChannel, channel.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table A.146: BusMirrorChannel

Class	***BusMirrorChannelMapping*** (abstract)			
Note	This element defines a bus mirroring in which the traffic from one communication bus (sourceChannel) is forwarded to another one (targetChannel). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	BusMirrorChannelMappingCan, BusMirrorChannelMappingFlexray, BusMirrorChannelMappingIp, BusMirrorChannelMappingUserDefined			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecuInstance	EcuInstance	0..1	ref	Ecu on which the BusMirroring is performed **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuInstance, ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
globalTime Domain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this BusMirrorChannelMapping shall be synchonized with. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTimeDomain.globalTimeDomain, globalTimeDomain.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
mirroring Protocol	MirroringProtocolEnum	0..1	attr	This attribute defines the bus mirroring protocol that is used in the BusMirrorChannelMapping

▽

△

Class	***BusMirrorChannelMapping*** (abstract)			
sourceChannel	BusMirrorChannel	0..1	aggr	Defines the sourceChannel from which frames are received. **Tags:** atp.Splitkey=sourceChannel
targetChannel	BusMirrorChannel	0..1	aggr	Defines the targetChannel to which frames are forwarded. **Tags:** atp.Splitkey=targetChannel
targetPdu Triggering	PduTriggering	*	ref	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=targetPduTriggering, targetPdu Triggering.pduTriggering, targetPduTriggering.variation Point.shortLabel vh.latestBindingTime=postBuild
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 A.147: BusMirrorChannelMapping

Class	**BusMirrorChannelMappingCan**			
Note	This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canIdRange Mapping	BusMirrorCanIdRange Mapping	*	aggr	Rules for remapping of a set of CAN IDs. **Tags:** atp.Splitkey=canIdRangeMapping
canIdToCanId Mapping	BusMirrorCanIdToCanId Mapping	*	aggr	Rules for remapping of single CanIds. **Tags:** atp.Splitkey=canIdToCanIdMapping
linPidToCanId Mapping	BusMirrorLinPidToCanIdMapping	*	aggr	Rules for remapping of single LIN Frames. **Tags:** atp.Splitkey=linPidToCanIdMapping
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 A.148: BusMirrorChannelMappingCan

Class	BusMirrorChannelMappingFlexray			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.149: BusMirrorChannelMappingFlexray

Class	BusMirrorChannelMappingIp			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.150: BusMirrorChannelMappingIp

Class	BusMirrorChannelMappingUserDefined			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a User Defined targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.151: BusMirrorChannelMappingUserDefined

Class	BusMirrorLinPidToCanIdMapping			
Note	This element defines a rule for remapping a single LIN Frame. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	BusMirrorChannelMappingCan.linPidToCanIdMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
remappedCanId	PositiveInteger	0..1	attr	This attribute defines the CanId on the targetChannel.

▽

△

Class	**BusMirrorLinPidToCanIdMapping**			
sourceLinPid	LinFrameTriggering	0..1	ref	This reference points to the sourceFrame with LIN identifier on the sourceChannel. **Tags:** atp.Splitkey=sourceLinPid

Table A.152: BusMirrorLinPidToCanIdMapping

Class	***BusspecificNmEcu*** (abstract)			
Note	Busspecific NmEcu attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	CanNmEcu, FlexrayNmEcu, J1939NmEcu, UdpNmEcu			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.153: BusspecificNmEcu

Enumeration	**ByteOrderEnum**
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.
Aggregated by	ApSomeipTransformationProps.byteOrder, BaseTypeDirectDefinition.byteOrder, DiagnosticCommonProps.defaultEndianness, ISignalToIPduMapping.packingByteOrder, MultiplexedIPdu.selectorFieldByteOrder, PduToFrameMapping.packingByteOrder, SegmentPosition.segmentByteOrder, SOMEIPTransformationDescription.byteOrder, System.containerIPduHeaderByteOrder
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 A.154: ByteOrderEnum

Class	**CalibrationParameterValue**			
Note	Specifies instance specific calibration parameter values used to initialize the memory objects implementing calibration parameters in the generated RTE code. RTE generator will use the implInitValue to override the initial values specified for the DataPrototypes of a component type. 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. Note: If both representations of init values are available these need to represent the same content. Note further that in this case an explicit mapping of ValueSpecification is not implemented because calibration parameters are delivered back after the calibration phase.			
Base	*ARObject*			
Aggregated by	CalibrationParameterValueSet.calibrationParameterValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ApplicationDataType **Tags:** atp.Splitkey=applInitValue
implInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ImplementationDataType **Tags:** atp.Splitkey=implInitValue
initialized Parameter	FlatInstanceDescriptor	0..1	ref	This represents the parameter that is initialized by the CalibrationParameterValue. **Tags:** atp.Splitkey=initializedParameter This Attribute is only used by the AUTOSAR Classic Platform.

Table A.155: CalibrationParameterValue

Enumeration	**CalprmAxisCategoryEnum**
Note	This enum specifies the possible values of the category property within SwCalprmAxis.
Aggregated by	RuleBasedAxisCont.category, SwAxisCont.category, SwCalprmAxis.category
Literal	***Description***
comAxis	COM_AXIS is equal to an STD_AXIS, the difference is, that a COM_AXIS is an shared axis, that means this axis can be used multiple times by different CURVEs, MAPs, CUBOIDs, CUBE_4s, and CUBE_5s. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=COM_AXIS
fixAXIS	FIX_AXIS means that the input axis is not stored. The axis is calculated using parameters and so on it is also not possible to modify the axis points. **Tags:** atp.EnumerationLiteralIndex=4 xml.name=FIX_AXIS
resAxis	RES_AXIS is also an shared axis like COM_AXIS, the difference is that this kind of axis can be used for rescaling. **Tags:** atp.EnumerationLiteralIndex=6 xml.name=RES_AXIS



△

Enumeration	**CalprmAxisCategoryEnum**
stdAxis	STD_AXIS means that input and output axis definition are stored within this CURVE, MAP, CUBOID, CUBE_4, and CUBE_5. There is no shared or calculated axis. **Tags:** atp.EnumerationLiteralIndex=8 xml.name=STD_AXIS

Table A.156: CalprmAxisCategoryEnum

Class	«atpVariation» **CanCluster**			
Note	CAN bus specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *AbstractCanCluster*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.157: CanCluster

Class	**CanCommunicationConnector**			
Note	CAN bus specific communication connector attributes.			
Base	*ARObject*, *AbstractCanCommunicationConnector*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
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 A.158: CanCommunicationConnector

Class	«atpVariation» **CanCommunicationController**			
Note	CAN bus specific communication port attributes.			
Base	*ARObject*, *AbstractCanCommunicationController*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.159: CanCommunicationController

Class	**CanControllerConfiguration**			
Note	This element is used for the specification of the exact CAN Bit Timing configuration parameter values.			
Base	*ARObject*, *AbstractCanCommunicationControllerAttributes*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes, CanXlProps.canConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
propSeg	Integer	0..1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	Integer	0..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	0..1	attr	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
timeSeg2	Integer	0..1	attr	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2

Table A.160: CanControllerConfiguration

Class	**CanControllerConfigurationRequirements**			
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*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	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.
minNumberOf TimeQuantaPer Bit	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.
minSyncJump Width	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 A.161: CanControllerConfigurationRequirements

Class	**CanControllerFdConfiguration**
Note	Bit timing related configuration of a CAN controller for payload and CRC of a CAN FD frame.
Base	*ARObject*
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdAttributes, CanXlProps.canFdConfig

▽

△

Class	**CanControllerFdConfiguration**			
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	0..1	attr	Specifies propagation delay in time quantas.
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	0..1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	0..1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	0..1	attr	Specifies phase segment 2 in time quantas.
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 A.162: CanControllerFdConfiguration

Class	**CanControllerFdConfigurationRequirements**			
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*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdRequirements			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	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.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	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.

▽

△

Class	**CanControllerFdConfigurationRequirements**			
minSyncJump Width	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.
minTrcvDelay Compensation Offset	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 A.163: CanControllerFdConfigurationRequirements

Class	**CanControllerXlConfiguration**			
Note	This meta-class represents the CAN XL-specific controller attributes.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlAttributes, CanXlProps.canXlConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
propSeg	PositiveInteger	0..1	attr	Specifies propagation delay in time quantas.
pwmL	PositiveInteger	0..1	attr	Specifies the PWM long phase length.
pwmO	PositiveInteger	0..1	attr	Specifies the PWM time offset.
pwmS	PositiveInteger	0..1	attr	Specifies the PWM short phase length.
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	0..1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	0..1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	0..1	attr	Specifies phase segment 2 in time quantas.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table A.164: CanControllerXlConfiguration

Class	**CanControllerXlConfigurationRequirements**			
Note	This element allows the specification of ranges for the CAN XL configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlRequirements, CanXlProps.canXlConfig Reqs			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxPwmL	PositiveInteger	0..1	attr	Specifies the maximum PWM long phase length.
maxPwmO	PositiveInteger	0..1	attr	Specifies the maximum PWM time offset.
maxPwmS	PositiveInteger	0..1	attr	Specifies the maximum PWM short phase length.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	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.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minPwmL	PositiveInteger	0..1	attr	Specifies the minimum PWM long phase length.
minPwmO	PositiveInteger	0..1	attr	Specifies the minimum PWM time offset.
minPwmS	PositiveInteger	0..1	attr	Specifies the minimum PWM short phase length.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJump Width	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.
minTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table A.165: CanControllerXlConfigurationRequirements

Class	**CanFrameTriggering**			
Note	CAN specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canAddressing Mode	CanAddressingMode Type	0..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	CanFrameRxBehavior Enum	0..1	attr	Defines which CAN protocol shall be expected for frame reception.
canFrameTx Behavior	CanFrameTxBehavior Enum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
canXlFrame TriggeringProps	CanXlFrameTriggering Props	0..1	aggr	Definition of CAN XL specific attributes in case the frame is a CAN XL frame. **Tags:** atp.Splitkey=canXlFrameTriggeringProps
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. **Tags:** atp.Splitkey=rxIdentifierRange
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 A.166: CanFrameTriggering

Enumeration	**CanFrameTxBehaviorEnum**
Note	Defines different CAN protocols for frame transmission behavior.
Aggregated by	CanFrameTriggering.canFrameTxBehavior, IEEE1722TpAcfCanPart.canFrameTxBehavior
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 A.167: CanFrameTxBehaviorEnum

Class	**CanNmCluster**			
Note	Can specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmBusload ReductionActive	Boolean	0..1	attr	It determines if bus load reduction for the respective Can Nm channel is active or not.
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering.



△

Class	**CanNmCluster**			
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.
nmImmediate NmCycleTime	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 CanNm ImmediateNmTransmissions is greater one.
nmImmediate Nm Transmissions	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.
nmMessage TimeoutTime	TimeValue	0..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.
nmMsgCycle Time	TimeValue	0..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.
nmNetwork Timeout	TimeValue	0..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.
nmPnHandle MultipleNetwork Requests	Boolean	0..1	attr	Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).
nmRemote SleepIndication Time	TimeValue	0..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.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	0..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 A.168: CanNmCluster

Class	**CanNmClusterCoupling**			
Note	CAN attributes that are valid for each of the referenced (coupled) CAN clusters. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *NmClusterCoupling*			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
coupledCluster	CanNmCluster	*	ref	Reference to coupled CAN Clusters. **Tags:** atp.Splitkey=coupledCluster
nmBusload Reduction Enabled	Boolean	0..1	attr	Enables busload reduction support

▽

△

Class	CanNmClusterCoupling			
nmImmediate RestartEnabled	Boolean	0..1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table A.169: CanNmClusterCoupling

Class	CanNmNode			
Note	CAN specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allNmMessages KeepAwake	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
nmCarWakeUp FilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported.
nmCarWakeUp RxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWake Up bit evaluation in received NmPdus.
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
nmMsg ReducedTime	TimeValue	0..1	attr	Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.

Table A.170: CanNmNode

Class	CanPhysicalChannel			
Note	CAN bus specific physical channel attributes.			
Base	*ARObject*, *AbstractCanPhysicalChannel*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.171: CanPhysicalChannel

Class	CanSecConnectivityAssociation			
Note	Definition of a Connectivity Association for CANsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**CanSecConnectivityAssociation**			
can Communication Connector	CanCommunication Connector	*	ref	Reference to the CanCommunicationConnectors this Can SecConnectivityAssociation is communicating on. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canCommunicationConnector, can CommunicationConnector.canCommunicationConnector, canCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild
secureChannel	CanSecSecureChannel	*	aggr	Collection of CanSecSecureChannels in the scope of this CanSecConnectivityAssociation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secureChannel, secureChannel.shortName, secureChannel.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.172: CanSecConnectivityAssociation

Class	**CanSecSecureChannel**			
Note	Definition of a CanSecSecureChannel.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanSecConnectivityAssociation.secureChannel			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
csci	PositiveInteger	0..1	attr	CANsec Secure Channel Identifier.
tx Communication Connector	CanCommunication Connector	0..1	ref	Reference to the transmitting communication connector. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=txCommunicationConnector, tx CommunicationConnector.canCommunicationConnector, txCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.173: CanSecSecureChannel

Class	**CanTpAddress**			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpAddress			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tpAddress	Integer	0..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 A.174: CanTpAddress

Class	CanTpChannel			
Note	Configuration parameters of the CanTp channel. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
channelId	PositiveInteger	0..1	attr	The id of the channel. The value shall be unique for each channel.

Table A.175: CanTpChannel

Class	CanTpConfig			
Note	This element defines exactly one CAN TP Configuration. One CanTpConfig element shall be created for each CAN Network in the System. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpAddress	CanTpAddress	*	aggr	Collection of TP Addresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpChannel	CanTpChannel	*	aggr	Configuration of CAN TP channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpChannel, tpChannel.shortName, tp Channel.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	CanTpConnection	*	aggr	Senders and receivers of CAN TP messages. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpEcu	CanTpEcu	*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpEcu, tpEcu.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**CanTpConfig**			
tpNode	CanTpNode	*	aggr	Senders and receivers of Can TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.176: CanTpConfig

Class	**CanTpConnection**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	CanTpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
addressing Format	CanTpAddressing FormatType	0..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	0..1	ref	Reference to the CanTpChannel on which this CanTp Connection is realized. **Tags:** atp.Splitkey=canTpChannel
dataPdu	NPdu	0..1	ref	Reference to an Data NPdu. **Tags:** atp.Splitkey=dataPdu
flowControlPdu	NPdu	0..1	ref	Reference to the Flow Control NPdu. **Tags:** atp.Splitkey=flowControlPdu
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. **Tags:** atp.Splitkey=multicast
padding Activation	Boolean	0..1	attr	This specifies whether 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. **Tags:** atp.Splitkey=receiver
taType	NetworkTargetAddress Type	0..1	attr	Network Target Address type.

▽

△

Class	**CanTpConnection**			
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.
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	0..1	ref	Reference to an IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=tpSdu
transmitter	CanTpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter

Table A.177: CanTpConnection

Class	**CanTpEcu**			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	CanTpConfig.tpEcu			
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	0..1	ref	Connection to the ECUInstance in the Topology **Tags:** atp.Splitkey=ecuInstance

Table A.178: CanTpEcu

Class	**CanTpNode**			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpNode			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
connector	Communication Connector	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). **Tags:** atp.Splitkey=connector

▽

△

Class	**CanTpNode**			
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). **Tags:** atp.Splitkey=tpAddress

Table A.179: CanTpNode

Class	**CanXlFrameTriggeringProps**			
Note	This element indicates the frame being CAN XL and contains further CAN XL specific attributes.			
Base	*ARObject*			
Aggregated by	CanFrameTriggering.canXlFrameTriggeringProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
acceptance Field	PositiveInteger	0..1	attr	Acceptance field of a CAN XL message.
priorityId	PositiveInteger	0..1	attr	Priority ID of a CAN XL message.
sduType	PositiveInteger	0..1	attr	SDU type of a CAN XL message.
vcid	PositiveInteger	0..1	attr	Virtual CAN network ID of a CAN XL message.

Table A.180: CanXlFrameTriggeringProps

Class	**Caption**			
Note	This meta-class represents the ability to express a caption which is a title, and a shortName.			
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	MlFigure.figureCaption, MlFormula.formulaCaption, Table.tableCaption			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
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! This property helps a human reader to identify the object in question. **Tags:** atp.Splitkey=desc xml.sequenceOffset=10

Table A.181: Caption

Class	**Chapter**			
Note	This meta-class represents a chapter of a document. Chapters are the primary structuring element in documentation.			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*			
Aggregated by	ChapterOrMsrQuery.chapter, MsrQueryResultChapter.chapter, SwComponentDocumentation.chapter, SwComponentDocumentation.swCalibrationNotes, SwComponentDocumentation.swCarbDoc, SwComponentDocumentation.swDiagnosticsNotes, SwComponentDocumentation.swFeatureDef, SwComponentDocumentation.swFeatureDesc, SwComponentDocumentation.swMaintenanceNotes, SwComponentDocumentation.swTestDesc, System.systemDocumentation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
chapterModel	ChapterModel	0..1	aggr	This represents the overall contents of the chapter. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.typeElement=false xml.typeWrapperElement=false
helpEntry	String	0..1	attr	This specifies an entry point in an online help system to be linked with the parent class. The syntax shall be defined by the applied help system respectively help system generator. Maybe it is a concatenated Identifier, but as of now we leave it as an arbitrary string. **Tags:** xml.attribute=true

Table A.182: Chapter

Class	**ClientComSpec**			
Note	Client-specific communication attributes (`RPortPrototype` typed by `ClientServerInterface`			
Base	*ARObject*, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
endToEndCall Response Timeout	TimeValue	0..1	attr	This attribute defines the maximum time interval in which the application shall expect the servers's response (time between the sending of the call invocation until the arrival of the server's response).
operation	ClientServerOperation	0..1	ref	This represents the corresponding ClientServerOperation. **Stereotypes:** atpIdentityContributor
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation. **Tags:** atp.Splitkey=transformationComSpecProps

Table A.183: ClientComSpec

Class	**ClientIdDefinition**			
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*			
Aggregated by	ClientIdDefinitionSet.clientIdDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ClientIdDefinition**			
clientId	Numerical	0..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	0..1	iref	Reference to the ClientServerOperation that is called by the client. **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef

Table A.184: ClientIdDefinition

Class	**ClientIdRange**			
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*			
Aggregated by	EcuInstance.clientIdRange, MachineDesign.clientIdRange			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
upperLimit	Limit	0..1	attr	This specifies the upper limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.185: ClientIdRange

Class	**ClientServerAnnotation**			
Note	Annotation to a port regarding a certain Operation.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.clientServerAnnotation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	ref	This represents the ClientServerOperation that the Client ServerAnnotation corresponds to. **Tags:** atp.Splitkey=operation

Table A.186: ClientServerAnnotation

Class	**ClientServerApplicationErrorMapping**			
Note	This meta-class represents the ability to map `ApplicationError`s onto each other.			
Base	*ARObject*			
Aggregated by	ClientServerInterfaceMapping.errorMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ClientServerApplicationErrorMapping**			
firstApplication Error	ApplicationError	0..1	ref	This represents the first ApplicationError in the context of the ClientServerApplicationErrorMapping. **Tags:** atp.Splitkey=firstApplicationError
second ApplicationError	ApplicationError	0..1	ref	This represents the second ApplicationError in the context of the ClientServerApplicationErrorMapping. **Tags:** atp.Splitkey=secondApplicationError

Table A.187: ClientServerApplicationErrorMapping

Class	**ClientServerInterface**			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	*	aggr	`ClientServerOperation`(s) of this `ClientServerInterface`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=operation, operation.shortName, operation.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface. **Tags:** atp.Splitkey=possibleError, possibleError.short Name

Table A.188: ClientServerInterface

Class	**ClientServerInterfaceMapping**			
Note	Defines the mapping of `ClientServerOperation`s in context of two different `ClientServerInterface`s.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
errorMapping	ClientServerApplication ErrorMapping	*	aggr	Map two different ApplicationErrors defined in the context of two different ClientServerInterfaces. **Tags:** atp.Splitkey=errorMapping
operation Mapping	ClientServerOperation Mapping	*	aggr	Mapping of two ClientServerOperations in two different ClientServerInterfaces **Tags:** atp.Splitkey=operationMapping

Table A.189: ClientServerInterfaceMapping

Class	ClientServerOperation			
Note	An operation declared within the scope of a client/server interface.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.command, *AtpClassifier*.atpFeature, ClientServerInterface.operation, DiagnosticDataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface.write, DiagnosticExtendedDataRecordInterface.provide, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutineInterface.stop, ServiceInterface.method			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this `ClientServerOperation`. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
diagArgIntegrity	Boolean	0..1	attr	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. 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. If the attribute does not exist or is set to false the `ClientServerOperation` does not have to consider the usage of a shared buffer. This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	ref	Possible errors that may by raised by the referring operation. **Tags:** atp.Splitkey=possibleError This Attribute is only used by the AUTOSAR Classic Platform.

Table A.190: ClientServerOperation

Class	ClientServerOperationMapping			
Note	Defines the mapping of two particular `ClientServerOperation`s in context of two different `ClientServerInterface`s.			
Base	*ARObject*			
Aggregated by	ClientServerInterfaceMapping.operationMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument Mapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular ArgumentData Prototypes with unequal names or unequal semantic (resolution or range) in context of Operations. **Tags:** atp.Splitkey=argumentMapping
firstOperation	ClientServerOperation	0..1	ref	First to-be-mapped ClientServerOperation of a Client ServerInterface. **Tags:** atp.Splitkey=firstOperation
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference indicates that a DataTransformation is intended in the context of the ClientServerOperation Mapping. **Tags:** atp.Splitkey=firstToSecondDataTransformation



△

Class	ClientServerOperationMapping			
second Operation	ClientServerOperation	0..1	ref	Second to-be-mapped ClientServerOperation of a Client ServerInterface. **Tags:** atp.Splitkey=secondOperation

Table A.191: ClientServerOperationMapping

Class	ClientServerToSignalMapping			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
callSignal	SystemSignal	0..1	ref	Reference to the callSignal to which the IN and INOUT ArgumentDataPrototypes are mapped. **Tags:** atp.Splitkey=callSignal
clientServer Operation	ClientServerOperation	0..1	iref	Reference to a ClientServerOperation, which is mapped to a call SystemSignal and a return SystemSignal. **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped. **Tags:** atp.Splitkey=returnSignal

Table A.192: ClientServerToSignalMapping

Class	Code			
Note	A generic code descriptor. The type of the code (source or object) is defined via the category attribute of the associated engineering object.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Implementation*.codeDescriptor			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
artifact Descriptor	AutosarEngineering Object	*	aggr	Refers to the artifact belonging to this code descriptor. **Tags:** atp.Splitkey=artifactDescriptor
callbackHeader	ServiceNeeds	*	ref	The association callbackHeader describes in which header files the function declarations of callback functions are provided to a service module. With this information the service module can include the appropriate header files in its configuration files. **Tags:** atp.Splitkey=callbackHeader

Table A.193: Code

Class	**Collection**			
Note	This meta-class specifies a collection of elements. A collection can be utilized to express additional aspects for a set of elements. Note that Collection is an ARElement. Therefore it is applicable e.g. for EvaluatedVariant, even if this is not obvious. Usually the category of a Collection is "SET". On the other hand, a Collection can also express an arbitrary relationship between elements. This is denoted by the category "RELATION" (see also [TPS_GST_00347]). In this case the collection represents an association from "sourceElement" to "targetElement" in the role "role". **Tags:** atp.recommendedPackage=Collections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
autoCollect	AutoCollectEnum	0..1	attr	This attribute reflects how far the referenced objects are part of the collection. **Tags:** xml.sequenceOffset=20
collected Instance	AtpFeature	*	iref	This instance ref supports the use case that a particular instance is part of the collection. **Tags:** atp.Splitkey=collectedInstance, collectedInstance.context Element, collectedInstance.target xml.sequenceOffset=60 **InstanceRef implemented by:** AnyInstanceRef
collection Semantics	NameToken	0..1	attr	Provides the ability to express the semantics of a Collection depending on the intended use case. The collectionSemantics is specified as a NameToken which must be agreed by all stakeholders. **Tags:** xml.sequenceOffset=25
element	Identifiable	*	ref	This is an element in the collection. Note that Collection itself is collectable. Therefore collections can be nested. In case of category="RELATION" this represents the target end of the relation. **Tags:** atp.Splitkey=element xml.sequenceOffset=40
elementRole	Identifier	0..1	attr	This attribute allows to denote a particular role of the collection. Note that the applicable semantics shall be mutually agreed between the two parties. In particular it denotes the role of element in the context of sourceElement. **Tags:** xml.sequenceOffset=30
sourceElement	Identifiable	*	ref	Only if Category = “RELATION”. This represents the source of a relation. **Tags:** atp.Splitkey=sourceElement xml.sequenceOffset=50
sourceInstance	AtpFeature	*	iref	Only if Category = “RELATION”. This represents the source instance of a relation. **Tags:** atp.Splitkey=sourceInstance, sourceInstance.context Element, sourceInstance.target xml.sequenceOffset=70 **InstanceRef implemented by:** AnyInstanceRef

Table A.194: Collection

Class	***CommConnectorPort*** (abstract)			
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	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	FramePort, IPduPort, ISignalPort			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..1	attr	Communication Direction of the Connector Port (input or output Port).

Table A.195: CommConnectorPort

Class	**CommunicationBufferLocking**			
Note	The aggregation of this meta-class specifies that a RunnableEntity supports locked communication buffers supplied by the RTE. It is able to cope with the error RTE_E_COM_BUSY.			
Base	*ARObject*, *SwcSupportedFeature*			
Aggregated by	PortAPIOption.supportedFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
supportBuffer Locking	SupportBufferLocking Enum	0..1	attr	This attribute is used to indicate the intended buffer locking behavior.

Table A.196: CommunicationBufferLocking

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
Note	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. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	*AbstractCanCluster*, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine.

▽

△

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
physical Channel	PhysicalChannel	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=physicalChannel, physicalChannel.shortName, physicalChannel.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

Table A.197: CommunicationCluster

Class	***CommunicationConnector*** (abstract)			
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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunicationConnector, LinCommunicationConnector, UserDefinedCommunicationConnector			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
commController	CommunicationController	0..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. **Tags:** atp.Splitkey=commController
createEcuWakeupSource	Boolean	0..1	attr	If this parameter is available and set to true then a channel wakeup source shall be created for the Physical Channel referencing this CommunicationConnector.
dynamicPncToChannelMappingEnabled	Boolean	0..1	attr	Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this CommunicationConnector and its respective Physical Channel. **Tags:** atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	***CommunicationConnector*** (abstract)			
ecuCommPort Instance	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:** atp.Splitkey=ecuCommPortInstance, ecuCommPort Instance.shortName, ecuCommPortInstance.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
explicitWakeup Channel	PhysicalChannel	*	ref	Defines a restriction which PhysicalChannels shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupChannel, explicitWakeup Channel.physicalChannel, explicitWakeup Channel.variationPoint.shortLabel vh.latestBindingTime=postBuild
explicitWakeup Pnc	PncMappingIdent	*	ref	Defines a restriction which PNCs shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupPnc, explicitWakeupPnc.pnc MappingIdent, explicitWakeupPnc.variationPoint.short Label vh.latestBindingTime=postBuild
nmVariant	NmVariantEnum	0..1	attr	Defines the functionality of Network Management.
pncFilterArray Mask (ordered)	PositiveInteger	*	attr	Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.
pncGateway Type	PncGatewayTypeEnum	0..1	attr	Defines if this EcuInstance shall implement the Pnc Gateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several Ecu Instances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active". This Attribute is only used by the AUTOSAR Classic Platform.

Table A.198: CommunicationConnector

Class	«atpVariation» ***CommunicationController*** (abstract)			
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, FlexrayCommunication Controller, *LinCommunicationController*, UserDefinedCommunicationController			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	«atpVariation» ***CommunicationController*** (abstract)			
wakeUpBy Controller Supported	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 A.199: CommunicationController

Class	**CommunicationControllerMapping**			
Note	CommunicationControllerMapping specifies the CommunicationPeripheral hardware (defined in the ECU Resource Template) to realize the specified CommunicationController in a physical topology.			
Base	*ARObject*			
Aggregated by	ECUMapping.commControllerMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Controller	Communication Controller	0..1	ref	Reference to the CommunicationController in the System Template **Tags:** atp.Splitkey=communicationController
hw Communication Controller	HwElement	0..1	ref	Reference to a HwElement of category Communication Controller in the ECU Resource Template. **Tags:** atp.Splitkey=hwCommunicationController

Table A.200: CommunicationControllerMapping

Class	***CommunicationCycle*** (abstract)			
Note	The communication cycle where the frame is sent.			
Base	*ARObject*			
Subclasses	CycleCounter, CycleRepetition			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.201: CommunicationCycle

Enumeration	**CommunicationDirectionType**
Note	Describes the communication direction.
Aggregated by	*CommConnectorPort*.communicationDirection, *IEEE1722TpConnection*.communicationDirection, IPSecRule.direction, ISignalIPduGroup.communicationDirection
Literal	***Description***
in	Reception (Input) **Tags:** atp.EnumerationLiteralIndex=0
out	Transmission (Output) **Tags:** atp.EnumerationLiteralIndex=1

Table A.202: CommunicationDirectionType

Class	ComplexDeviceDriverSwComponentType			
Note	The ComplexDeviceDriverSwComponentType is a special AtomicSwComponentType that has direct access to hardware on an ECU and which is therefore linked to a specific ECU or specific hardware. The ComplexDeviceDriverSwComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hardware Element	HwDescriptionEntity	*	ref	Reference from the ComplexDeviceDriverSwComponent Type to the description of the used HwElements. **Tags:** atp.Splitkey=hardwareElement

Table A.203: ComplexDeviceDriverSwComponentType

Class	ComponentClustering			
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*			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clustered Component	SwComponent Prototype	*	iref	Reference to the components that have to be mapped together. **Tags:** atp.Splitkey=clusteredComponent, clustered Component.contextComposition, clustered Component.contextComponent, clustered Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
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 mappingScope Ecu shall be assumed.

Table A.204: ComponentClustering

Class	ComponentSeparation			
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.			
Base	*ARObject*, *MappingConstraint*			
Aggregated by	SystemMapping.mappingConstraint			
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.

▽

△

Class	**ComponentSeparation**			
separated Component	SwComponent Prototype	0..2	iref	The two components that have to be mapped to different ECUs **Tags:** atp.Splitkey=separatedComponent, separated Component.contextComposition, separated Component.contextComponent, separated Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table A.205: ComponentSeparation

Class	**CompositeNetworkRepresentation**			
Note	This meta-class is used to define the network representation of leaf elements of composite application data types.			
Base	*ARObject*			
Aggregated by	*ReceiverComSpec*.compositeNetworkRepresentation, *SenderComSpec*.compositeNetwork Representation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
leafElement	ApplicationComposite ElementDataPrototype	0..1	iref	This represents that leaf element of an application composite data type. **Tags:** atp.Splitkey=leafElement, leafElement.rootData Prototype, leafElement.contextDataPrototype, leaf Element.targetDataPrototype **InstanceRef implemented by:** ApplicationComposite ElementInPortInterfaceInstanceRef
network Representation	SwDataDefProps	0..1	aggr	The SwDataDefProps owned by the CompositeNetwork Representation are used to define the network representation of the leaf element of an Application CompositeDataType. **Tags:** atp.Splitkey=networkRepresentation

Table A.206: CompositeNetworkRepresentation

Class	**CompositeRuleBasedValueSpecification**			
Note	This meta-class represents rule-based values for DataPrototypes typed by composite AutosarDataTypes.			
Base	*ARObject*, *AbstractRuleBasedValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	CompositeRuleBasedValueSpecification			
argument (ordered)	CompositeValue Specification	*	aggr	This represents the collection of aggregated Value Specifications. The last ValueSpecification in the collection shall be taken to execute the filling rule. **Tags:** xml.sequenceOffset=30
compound Primitive Argument (ordered)	CompositeRuleBased ValueArgument	*	aggr	This represents the collection of aggregated Value Specifications for compound primitive data type. The last ValueSpecification in the collection shall be taken to execute the filling rule. **Tags:** xml.sequenceOffset=35
maxSizeToFill	PositiveInteger	0..1	attr	If a rule is chosen which does not fill until the end, this determines until which size the rule shall fill the values. **Tags:** xml.sequenceOffset=40
rule	Identifier	0..1	attr	This denotes the name of the rule of the RuleBasedValue Specification. The rule determines the calculation specification according which the arguments are used to calculated the values. **Tags:** xml.sequenceOffset=20

Table A.207: CompositeRuleBasedValueSpecification

Class	*CompositeValueSpecification* (abstract)			
Note	This abstract meta-class acts a base class for ValueSpecifications that have a composite form.			
Base	*ARObject*, *ValueSpecification*			
Subclasses	ArrayValueSpecification, RecordValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.argument, Constant Specification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compare Value, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compare Value, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.208: CompositeValueSpecification

Class	CompositionSwComponentType			
Note	A `CompositionSwComponentType` aggregates `SwComponentPrototype`s (that in turn are typed by `SwComponentType`)s as well as `SwConnector`s for primarily connecting `SwComponentPrototype`s among each others and towards the surface of the `CompositionSwComponentType`. By this means, a hierarchical structures of software-components can be created. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**CompositionSwComponentType**			
component	SwComponent Prototype	*	aggr	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. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=component, component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	SwConnector	*	aggr	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. The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec. **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMappingSet to be applied for the used ApplicationDataTypes in PortInterfaces. 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 CompositionSwComponentTypes. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility. **Tags:** atp.Splitkey=dataTypeMapping

▽

△

Class	**CompositionSwComponentType**			
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationRTEEventProps, instantiation RTEEventProps.shortLabel, instantiationRTEEvent Props.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime This Attribute is only used by the AUTOSAR Classic Platform.
physical Dimension Mapping	PhysicalDimension MappingSet	0..1	ref	This reference identifies the `PhysicalDimensionMappingSet` that is applicable in the context of the enclosing `CompositionSwComponentType`. The `PhysicalDimensionMapping`s contained in the `PhysicalDimensionMappingSet` shall be taken into account for the assessment of the compatibility of `PhysicalDimension`s in the context of creation of a `PortInterfaceMapping` in the scope of the `CompositionSwComponentType`. **Tags:** atp.Splitkey=physicalDimensionMapping

Table A.209: CompositionSwComponentType

Class	**Compu**			
Note	This meta-class represents the ability to express one particular computation.			
Base	*ARObject*			
Aggregated by	CompuMethod.compuInternalToPhys, CompuMethod.compuPhysToInternal			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuContent	CompuContent	0..1	aggr	This specifies the details of the computation. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
compuDefault Value	CompuConst	0..1	aggr	This property can be used to specify an output value for a conversion formula, if the value to be converted lies outside the plausibility limit. Although this is possible for all conversion formulae, it is especially valid for variables with tabular conversion formulae. **Tags:** atp.Splitkey=compuDefaultValue xml.sequenceOffset=70

Table A.210: Compu

Class	**CompuConst**			
Note	This meta-class represents the fact that the value of a computation method scale is constant.			
Base	*ARObject*			
Aggregated by	Compu.compuDefaultValue, CompuScale.compuInverseValue, CompuScaleConstantContents.compu Const			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	CompuConst			
compuConst ContentType	CompuConstContent	0..1	aggr	This is the actual content of the constant compu method scale. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=10 xml.typeElement=false xml.typeWrapperElement=false

Table A.211: CompuConst

Class	CompuConstFormulaContent			
Note	This meta-class represents the fact that the constant value of the computation method is represented by a variation point. This difference is due to compatibility with ASAM HDO.			
Base	*ARObject*, *CompuConstContent*			
Aggregated by	CompuConst.compuConstContentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
vf	Numerical	0..1	attr	Value calculated via a system constant. This element is included in every case where parameters should be generated from numerical values during compile time (not runtime!). Thus for example, the influence of the cylinder number on conversion formulae can be introduced in a repeatable manner. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=30

Table A.212: CompuConstFormulaContent

Class	CompuConstTextContent			
Note	This meta-class represents the textual content of a scale.			
Base	*ARObject*, *CompuConstContent*			
Aggregated by	CompuConst.compuConstContentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
vt	VerbatimString	0..1	attr	This represents a textual constant in the computation method.

Table A.213: CompuConstTextContent

Class	CompuMethod
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. 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. **Tags:** atp.recommendedPackage=CompuMethods
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**CompuMethod**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuInternal ToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** atp.Splitkey=compuInternalToPhys xml.sequenceOffset=80
compuPhysTo Internal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:** atp.Splitkey=compuPhysToInternal 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:** atp.Splitkey=unit xml.sequenceOffset=30

Table A.214: CompuMethod

Class	**CompuNominatorDenominator**			
Note	This class represents the ability to express a polynomial either as Nominator or as Denominator.			
Base	*ARObject*			
Aggregated by	CompuRationalCoeffs.compuDenominator, CompuRationalCoeffs.compuNumerator			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
v (ordered)	Numerical	*	attr	this is the list of polynomial factors. Note that the first vf represents the power=0. The polynomial is v[0] * x^0 + v[1] * x^1 ... **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.215: CompuNominatorDenominator

Class	**CompuRationalCoeffs**			
Note	This meta-class represents the ability to express a rational function by specifying the coefficients of nominator and denominator.			
Base	*ARObject*			
Aggregated by	CompuScaleRationalFormula.compuRationalCoeffs			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compu Denominator	CompuNominator Denominator	0..1	aggr	This is the denominator of the expression. **Tags:** xml.sequenceOffset=30
compu Numerator	CompuNominator Denominator	0..1	aggr	This is the numerator of the rational expression. **Tags:** xml.sequenceOffset=20

Table A.216: CompuRationalCoeffs

Class	**CompuScale**			
Note	This meta-class represents the ability to specify one segment of a segmented computation method.			
Base	*ARObject*			
Aggregated by	CompuScales.compuScale			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
a2lDisplayText	String	0..1	attr	The value of this attribute shall be taken for generating one display text (specifically the OutVal) within the equivalent of the enclosing `CompuMethod` in A2L.
compuInverse Value	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:** atp.Splitkey=compuInverseValue xml.sequenceOffset=60
compuScale Contents	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	MultiLanguageOverview Paragraph	0..1	aggr	<desc> represents a general but brief description of the object in question. **Tags:** atp.Splitkey=desc 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
mask	PositiveUnlimitedInteger	0..1	attr	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. 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. The processing has to be done in order of the COMPU-SCALE elements. **Tags:** xml.sequenceOffset=35
shortLabel	Identifier	0..1	attr	This element specifies a short name for the particular scale. The name can for example be used to derive a programming language identifier. **Tags:** xml.sequenceOffset=20
symbol	CIdentifier	0..1	attr	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. **Tags:** xml.sequenceOffset=25
upperLimit	Limit	0..1	attr	This specifies the upper limit of a of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

Table A.217: CompuScale

Class	CompuScaleConstantContents			
Note	This meta-class represents the fact that a particular scale of the computation method is constant.			
Base	*ARObject*, *CompuScaleContents*			
Aggregated by	CompuScale.compuScaleContents			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuConst	CompuConst	0..1	aggr	This represents the fact that the scale is a constant. The use case is mainly a non interpolated scale. It is a simplification of the fact that a constant scale can also be expressed as rational function of order 0. **Tags:** xml.sequenceOffset=90

Table A.218: CompuScaleConstantContents

Class	CompuScales			
Note	This meta-class represents the ability to stepwise express a computation method.			
Base	*ARObject*, *CompuContent*			
Aggregated by	Compu.compuContent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuScale (ordered)	CompuScale	*	aggr	This represents one scale within the compu method. Note that it contains a Variationpoint in order to support blueprints of enumerations. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table A.219: CompuScales

Class	ConcretePatternEventTriggering			
Note	Describes the behavior of an event that occurs according to a precisely known pattern.			
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Timing Constraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
offset	MultidimensionalTime	*	aggr	The offset for each occurrence of the event in the specified time interval. A list of point-in-times in the time interval given by the parameter patternLength at which the event occurs. **Tags:** atp.Splitkey=offset xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=10 xml.typeElement=false
patternJitter	MultidimensionalTime	0..1	aggr	The maximum deviation of the time interval's starting point from the beginning of the given period. This parameter is only applicable in conjunction with the parameter `patternPeriod`. **Tags:** atp.Splitkey=patternJitter

△

Class	**ConcretePatternEventTriggering**			
patternLength	MultidimensionalTime	0..1	aggr	The duration of the time interval within which the event repeatedly occurs. The event occurs at concrete points in time within the given time interval. **Tags:** atp.Splitkey=patternLength xml.sequenceOffset=20
patternPeriod	MultidimensionalTime	0..1	aggr	The time distance between the beginnings of subsequent repetitions of the given concrete pattern. **Tags:** atp.Splitkey=patternPeriod

Table A.220: ConcretePatternEventTriggering

Class	«atpMixedString» **ConditionByFormula**			
Note	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. The result of the expression is interpreted as a condition. • "0" represents "false"; • a value other than zero is considered "true"			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Aggregated by	VariationPoint.swSyscond, VariationPointProxy.conditionAccess			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bindingTime	BindingTimeEnum	0..1	attr	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. **Tags:** xml.attribute=true

Table A.221: ConditionByFormula

Class	**ConditionalChangeNad**			
Note	Generates an conditional change NAD request. See ISO 17987 protocol specification for more information. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
byte	Integer	0..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	0..1	attr	Byte Position of Id.
invert	Integer	0..1	attr	Byte Position of Invert.
mask	Integer	0..1	attr	Byte Position of Mask.
newNad	Integer	0..1	attr	The newly assigned NAD value (Byte Position).

Table A.222: ConditionalChangeNad

Class	**ConfidenceInterval**
Note	Additionally to the list of measured distances of event occurrences, a confidence interval can be specified for the expected distance of two consecutive event occurrences with a given probability.
Base	*ARObject*

▽

△

Class	**ConfidenceInterval**			
Aggregated by	ArbitraryEventTriggering.confidenceInterval			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerBound	MultidimensionalTime	0..1	aggr	The lower bound of the expected distance of two consecutive event occurrences. **Tags:** atp.Splitkey=lowerBound
propability	Float	0..1	attr	The probability for the measured lower and upper bound of the confidence interval.
upperBound	MultidimensionalTime	0..1	aggr	The upper bound of the expected distance of two consecutive event occurrences. **Tags:** atp.Splitkey=upperBound

Table A.223: ConfidenceInterval

Class	**ConsistencyNeeds**			
Note	This meta-class represents the ability to define requirements on the implicit communication behavior.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ConsistencyNeedsBlueprintSet.consistencyNeeds, *SwComponentType*.consistencyNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dpgDoesNot Require Coherency	DataPrototypeGroup	*	aggr	This group of VariableDataPrototypes does not require coherency with respect to the implicit communication behavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dpgDoesNotRequireCoherency, dpgDoes NotRequireCoherency.shortName, dpgDoesNotRequire Coherency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dpgRequires Coherency	DataPrototypeGroup	*	aggr	This group of VariableDataPrototypes requires coherency with respect to the implicit communication behavior, i.e. all read and write access to VariableDataPrototypes in the DataPrototypeGroup by the RunnableEntitys of the RunnableEntityGroup need to be handled in a coherent manner. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dpgRequiresCoherency, dpgRequires Coherency.shortName, dpgRequiresCoherency.variation Point.shortLabel vh.latestBindingTime=preCompileTime
regDoesNot RequireStability	RunnableEntityGroup	*	aggr	This group of RunnableEntities does not require stability with respect to the implicit communication behavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=regDoesNotRequireStability, regDoesNot RequireStability.shortName, regDoesNotRequire Stability.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**ConsistencyNeeds**			
regRequires Stability	RunnableEntityGroup	*	aggr	This group of RunnableEntities requires stability with respect to the implicit communication behavior, i.e. all read and write access to VariableDataPrototypes in the DataPrototypeGroup by the RunnableEntitys of the RunnableEntityGroup need to be handled in a stable manner. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=regRequiresStability, regRequires Stability.shortName, regRequiresStability.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.224: ConsistencyNeeds

Class	**ConstantReference**			
Note	Instead of defining this value inline, a constant is referenced.			
Base	*ARObject*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
constant	ConstantSpecification	0..1	ref	The referenced constant. **Tags:** atp.Splitkey=constant

Table A.225: ConstantReference

Class	**ConstantSpecification**			
Note	Specification of a constant that can be part of a package, i.e. it can be defined stand-alone. **Tags:** atp.recommendedPackage=ConstantSpecifications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
valueSpec	ValueSpecification	0..1	aggr	Specification of an expression leading to a value for this constant. **Tags:** atp.Splitkey=valueSpec

Table A.226: ConstantSpecification

Class	ConstantSpecificationMapping			
Note	This meta-class is used to create an association of two ConstantSpecifications. One Constant Specification is supposed to be defined in the application domain while the other should be defined in the implementation domain. Hence the ConstantSpecificationMapping needs to be used where a ConstantSpecification defined in one domain needs to be associated to a ConstantSpecification in the other domain. This information is crucial for the RTE generator.			
Base	*ARObject*			
Aggregated by	ConstantSpecificationMappingSet.mapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
applConstant	ConstantSpecification	0..1	ref	A ConstantSpecification defined in the application domain. **Tags:** atp.Splitkey=applConstant
implConstant	ConstantSpecification	0..1	ref	A ConstantSpecification defined in the implementation domain. **Tags:** atp.Splitkey=implConstant

Table A.227: ConstantSpecificationMapping

Class	ConstantSpecificationMappingSet			
Note	This meta-class represents the ability to map two ConstantSpecifications to each others. One Constant Specification is supposed to be described in the application domain and the other should be described in the implementation domain. **Tags:** atp.recommendedPackage=ConstantSpecificationMappingSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mapping	ConstantSpecification Mapping	*	aggr	ConstantSpecificationMappings owned by the Constant SpecificationMappingSet. **Tags:** atp.Splitkey=mapping

Table A.228: ConstantSpecificationMappingSet

Class	ConsumedEventGroup			
Note	This element represents an event-group to which the service consumer wants to subscribe.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ConsumedServiceInstance.consumedEventGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autoRequire	Boolean	0..1	attr	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 ConsumedService Instance.autoRequire is set to FALSE.
eventGroup Identifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.



△

Class	**ConsumedEventGroup**			
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 TcpIp 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 ConsumedEventGroup shall point to different ApplicationEndpoint objects to reserve the necessary resources in the configuration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastAddress, eventMulticast Address.applicationEndpoint, eventMulticast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events. **Tags:** atp.Splitkey=pduActivationRoutingGroup, pdu ActivationRoutingGroup.shortName
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
sdClientTimer Config	SomeipSdClientEvent GroupTimingConfig	0..1	ref	Client Timing configuration settings that are EventGroup specific. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig, sdClientTimer Config.someipSdClientEventGroupTimingConfig, sdClient TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.229: ConsumedEventGroup

Class	**ConsumedServiceInstance**			
Note	Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ConsumedServiceInstance**			
allowedService Provider	NetworkEndpoint	*	ref	NetworkEndpoint on which the ProvidedServiceInstance that is communicating with this ConsumedService Instance is allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=allowedServiceProvider, allowedService Provider.networkEndpoint, allowedService Provider.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
autoRequire	Boolean	0..1	attr	Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.
blocklisted Version	SomeipServiceVersion	*	aggr	Collection of blocklisted versions **Tags:** atp.Splitkey=blocklistedVersion atp.Status=draft
consumedEvent Group	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedEventGroup, consumedEvent Group.shortName, consumedEventGroup.variation Point.shortLabel vh.latestBindingTime=postBuild
eventMulticast Subscription Address	ApplicationEndpoint	0..1	ref	Multicast Address that is used by the client to subscribe to the server: This enables the multicast subscription feature. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastSubscriptionAddress, event MulticastSubscriptionAddress.applicationEndpoint, event MulticastSubscriptionAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	AnyServiceInstanceId	0..1	attr	This attribute represents the ability to describe the required service instance ID.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the CSI is consumed (udp, tcp or both). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel 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.

▽

△

Class	**ConsumedServiceInstance**			
remoteUnicast Address	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:** atp.Splitkey=remoteUnicastAddress, remoteUnicast Address.applicationEndpoint, remoteUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdClientTimer Config	SomeipSdClientService InstanceConfig	0..1	ref	Client specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig, sdClientTimer Config.someipSdClientServiceInstanceConfig, sdClient TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is searched.
versionDriven FindBehavior	ServiceVersion AcceptanceKindEnum	0..1	attr	Defines the service discovery find behavior. **Tags:** atp.Status=draft

Table A.230: ConsumedServiceInstance

Enumeration	**ContainedIPduCollectionSemanticsEnum**
Note	Defines the collection semantics for ContainedIPdus.
Aggregated by	ContainedIPduProps.collectionSemantics
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 A.231: ContainedIPduCollectionSemanticsEnum

Class	**ContainedIPduProps**			
Note	Defines the aspects of an IPdu which can be collected inside a ContainerIPdu.			
Base	*ARObject*			
Aggregated by	ContainerIPdu.containedIPduTriggeringProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
collection Semantics	ContainedIPdu CollectionSemantics Enum	0..1	attr	Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.
containedPdu Triggering	PduTriggering	0..1	ref	Reference to Pdu for which the ContainedIPduProps are valid. **Tags:** atp.Splitkey=containedPduTriggering

▽

△

Class	**ContainedIPduProps**			
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.
j1939requestable	Boolean	0..1	attr	Contained I-PDU can be triggered by the J1939 request message.
offset	PositiveInteger	0..1	attr	Byte offset that describes the location of the Contained Pdu in the ContainerPdu if no header is used.
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 an IPdu specific sender timeout which adapts the ContainerIPdu transmission timer when this contained IPdu is put inside the ContainerIPdu: • f the ContainerIPdu transmission timer is already running, and the transmission timer value is greater than ContainedIPduProps.timeout, then the transmission timer value is reduced to ContainedIPdu Props.timeout. • If the ContainerIPdu transmission timer is already running, and the transmission timer value is less than or equal to ContainedIPduProps.timeout, then the transmission timer value is not changed. • If the ContainerIPdu transmission timer is not running, then the transmission timer value is set to Contained IPduProps.timeout. This attribute is ignored on receiver side.
trigger	PduCollectionTrigger Enum	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.

Table A.232: ContainedIPduProps

Class	**ContainerIPdu**			
Note	Allows to collect several IPdus in one ContainerIPdu based on the headerType. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
containedIPdu TriggeringProps	ContainedIPduProps	*	aggr	Defines properties for an IPdu that is part of the ContainerIPdu. **Tags:** atp.Splitkey=containedIPduTriggeringProps
container Timeout	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.

▽

△

Class	**ContainerIPdu**			
containerTrigger	ContainerIPduTrigger Enum	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	ContainerIPduHeader TypeEnum	0..1	attr	Defines whether and which header type is used (header id and length).
minimumRx Container QueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for received containers.
minimumTx Container QueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for transmitted containers.
rxAccept ContainedIPdu	RxAcceptContained IPduEnum	0..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.
unusedBit Pattern	PositiveInteger	0..1	attr	IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Table A.233: ContainerIPdu

Enumeration	**ContainerIPduHeaderTypeEnum**
Note	Is used to define the header type and size of ContainerIPdus. The header size includes the header id and the length information.
Aggregated by	ContainerIPdu.headerType
Literal	*Description*
longHeader	Header size is 64 bit: • 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
shortHeader	Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit. **Tags:** atp.EnumerationLiteralIndex=1

Table A.234: ContainerIPduHeaderTypeEnum

Class	**CouplingElement**			
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*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**CouplingElement**			
communication Cluster	EthernetCluster	0..1	ref	This relationship defines to which cluster the Coupling Element belongs. **Tags:** atp.Splitkey=communicationCluster
coupling ElementDetails	CouplingElement AbstractDetails	0..1	aggr	Definition of details for this specific CouplingElement. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingElementDetails, couplingElement Details.shortName, couplingElementDetails.variation Point.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild xml.namePlural=COUPLING-ELEMENT-DETAILS This Attribute is only used by the AUTOSAR Classic Platform.
couplingPort	CouplingPort	*	aggr	Hardware Port of the CouplingElement that is used to connect this CouplingPort to EcuInstances or other CouplingElements. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPort, couplingPort.shortName, couplingPort.variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingType	CouplingElementEnum	0..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. **Tags:** atp.Splitkey=ecuInstance
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of a Coupling Element. **Tags:** atp.Splitkey=firewallRule
switchMac Address LearningMode	SwitchMacAddress LearningEnum	0..1	attr	Defines the MAC address learning mode of the Ethernet switch.

Table A.235: CouplingElement

Enumeration	**CouplingElementEnum**
Note	Identifies the Coupling type.
Aggregated by	CouplingElement.couplingType
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 A.236: CouplingElementEnum

Class	**CouplingElementSwitchDetails**			
Note	Collection of specific details for the CouplingElement of couplingType switch. **Tags:** atp.Status=candidate atp.recommendedPackage=SwitchStreamIdentificationTables This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CouplingElementAbstractDetails*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElement.couplingElementDetails			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atsInstance Entry	SwitchAtsInstanceEntry	*	aggr	Collection of ATS Instance Entries. **Tags:** atp.Splitkey=atsInstanceEntry, atsInstanceEntry.short Name atp.Status=candidate
flowMetering	SwitchFlowMetering Entry	*	aggr	Collection of Flow Metering Entries. **Tags:** atp.Splitkey=flowMetering, flowMetering.shortName atp.Status=candidate
streamFilter (ordered)	SwitchStreamFilterEntry	*	aggr	Collection of Stream Filter Entries. **Tags:** atp.Status=candidate
streamGate	SwitchStreamGateEntry	*	aggr	Collection of Stream Gate Entries. **Tags:** atp.Splitkey=streamGate, streamGate.shortName atp.Status=candidate
switchStream Identification (ordered)	SwitchStream Identification	*	aggr	Collection of switch stream identification entries. **Tags:** atp.Status=candidate
trafficShaper Group	SwitchAsynchronous TrafficShaperGroup Entry	*	aggr	Collection of Traffic Shaper Groups. **Tags:** atp.Splitkey=trafficShaperGroup, trafficShaper Group.shortName atp.Status=candidate

Table A.237: CouplingElementSwitchDetails

Class	**CouplingPort**			
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*			
Aggregated by	CouplingElement.couplingPort, EthernetCommunicationController.couplingPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connection Negotiation Behavior	EthernetConnection NegotiationEnum	0..1	attr	Specifies the connection negotiation of the CouplingPort.
couplingPort Details	CouplingPortDetails	0..1	aggr	Defines more details of a CouplingPort in case a more specific configuration is required. **Tags:** atp.Splitkey=couplingPortDetails
couplingPort Role	CouplingPortRoleEnum	0..1	attr	Defines the role this CouplingPort takes in the context of the CouplingElement.

△

Class	**CouplingPort**			
defaultVlan	EthernetPhysical Channel	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). **Tags:** atp.Splitkey=defaultVlan
macAddress VlanAssignment	MacAddressVlan Membership	*	aggr	Statically defines the assignment of MAC-Multicast-Addresses, optionally together with VLANs, to this CouplingPort. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=macAddressVlanAssignment, macAddress VlanAssignment.shortName, macAddressVlan Assignment.variationPoint.shortLabel vh.latestBindingTime=postBuild
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the CouplingPort.
macSecProps	MacSecProps	*	aggr	Properties to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:** atp.Splitkey=macSecProps atp.Status=candidate
physicalLayer Type	EthernetPhysicalLayer TypeEnum	0..1	attr	Specifies the physical layer type of the CouplingPort.
plcaProps	PlcaProps	0..1	aggr	Optional properties for configuration of PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the Coupling Port (PHY). **Tags:** atp.Splitkey=plcaProps
pncMapping	PncMappingIdent	*	ref	Reference to the partial networks this CouplingPort participates in. **Tags:** atp.Splitkey=pncMapping
receiveActivity	EthernetSwitchVlan IngressTagEnum	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. **Tags:** atp.Splitkey=vlanMembership
wakeupSleep OnDataline Config	EthernetWakeupSleep OnDatalineConfig	0..1	ref	Optional reference to EthernetWakeupSleepOnDataline Config. **Tags:** atp.Splitkey=wakeupSleepOnDatalineConfig

Table A.238: CouplingPort

Class	**CouplingPortConnection**			
Note	Connection between two CouplingPorts (firstPort and secondPort) or between a collection of Ports that are all referenced by the portCollection reference.			
Base	*ARObject*			
Aggregated by	EthernetCluster.couplingPortConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstPort	CouplingPort	0..1	ref	Reference to the first CouplingPort that is connected via the CouplingPortConnection. **Tags:** atp.Splitkey=firstPort
nodePort	CouplingPort	*	ref	Reference to a number of CouplingPorts that are connected via the CouplingPortConnection. This reference shall be used to describe a 10BASE-T1S topology architecture where several CouplingPorts of EthernetCommunicationControllers are connected via one CouplingPortConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nodePort, nodePort.couplingPort, node Port.variationPoint.shortLabel vh.latestBindingTime=postBuild
plcaLocalNode Count	PositiveInteger	0..1	attr	Defines the number of communication participants in case 10BASE-T1S and the nodePort reference is used.
plcaTransmit Opportunity Timer	PositiveInteger	0..1	attr	Timer for the transmission in bit time to evaluate if a Transmission Opportunity is yield or not.
secondPort	CouplingPort	0..1	ref	Reference to the second CouplingPort that is connected via the CouplingPortConnection. **Tags:** atp.Splitkey=secondPort

Table A.239: CouplingPortConnection

Class	**CouplingPortDetails**			
Note	Defines details of a CouplingPort. May be used to configure the structures of a switch.			
Base	*ARObject*			
Aggregated by	CouplingPort.couplingPortDetails			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort Structural Element	CouplingPortStructural Element	*	aggr	Collects all the structural parts at which a CouplingPort may be configurable. **Tags:** atp.Splitkey=couplingPortStructuralElement, couplingPortStructuralElement.shortName
defaultTraffic Class	PositiveInteger	0..1	attr	Defines the default traffic class for this CouplingPort.
ethernetPriority Regeneration	EthernetPriority Regeneration	0..8	aggr	Defines a priority regeneration where the ingress priority is replaced by regenerated priority. **Tags:** atp.Splitkey=ethernetPriorityRegeneration, ethernetPriorityRegeneration.shortName
ethernetTraffic Class Assignment	CouplingPortTraffic ClassAssignment	*	aggr	Defines the priority to traffic class assignment. **Tags:** atp.Splitkey=ethernetTrafficClassAssignment, ethernetTrafficClassAssignment.shortName
frame Preemption Support	Boolean	0..1	attr	Defines whether frames handled by this CouplingPort may be preemted.

▽

△

Class	CouplingPortDetails			
globalTime Props	GlobalTimeCoupling PortProps	0..1	aggr	Specifies properties for the usage of the CouplingPort in the scope of Global Time Sync. **Tags:** atp.Splitkey=globalTimeProps
lastEgress Scheduler	CouplingPortScheduler	0..1	ref	Defines which CouplingPortScheduler is the last in the egress port structure. **Tags:** atp.Splitkey=lastEgressScheduler
ratePolicy	CouplingPortRatePolicy	*	aggr	Rate policies to be applied for this CouplingPort. **Tags:** atp.Splitkey=ratePolicy atp.Status=obsolete
vlanTranslation Table	EthernetVlanTranslation Table	*	aggr	Definition of entries that define the ingress Vlan translation between IngressVlanID and TranslatedVlanID. **Tags:** atp.Splitkey=vlanTranslationTable

Table A.240: CouplingPortDetails

Class	CouplingPortEnhancedTrafficShaper			
Note	Defines a scheduler used for enhanced traffic shaping (e.g. weighted round robin). **Tags:** atp.Status=candidate			
Base	*ARObject*, *CouplingPortAbstractShaper*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortFifo.shaper			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
etsAvailable BandwidthIn Percent	PositiveInteger	0..1	attr	Defines the available bandwidth in percent of an enhanced transmission selection algorithm (ETS). **Tags:** atp.Status=candidate
etsAvailable BandwidthIn WeightValue	PositiveInteger	0..1	attr	Defines the available bandwidth as weight value of an enhanced transmission selection algorithm (ETS). **Tags:** atp.Status=candidate

Table A.241: CouplingPortEnhancedTrafficShaper

Class	CouplingPortFifo			
Note	Defines a FIFO for the CouplingPort egress structure.			
Base	*ARObject*, *CouplingPortStructuralElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortDetails.couplingPortStructuralElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
assignedTraffic Class	PositiveInteger	*	attr	Defines a set of Traffic Classes which shall be handled by this FIFO.
minimumFifo Length	PositiveInteger	0..1	attr	FIFO minimum length in Byte. An actual configuration/ hardware may use a bigger value.
shaper	CouplingPortAbstract Shaper	0..1	aggr	Definition of the shaper to be used for the processing of this FIFO. **Tags:** atp.Splitkey=shaper, shaper.shortName atp.Status=candidate
trafficClass Preemption Support	EthernetCouplingPort PreemptionEnum	0..1	attr	Defines whether frames assigned to the traffic class associated with this CouplingPortFifo may be preempted or not.

Table A.242: CouplingPortFifo

Class	**CouplingPortRatePolicy**			
Note	Defines a rate policy on a CouplingPort. **Tags:** atp.Status=obsolete			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.ratePolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	Amount of data in bytes (excluding header information) that can be received to define the rate policy. **Tags:** atp.Status=obsolete
policyAction	CouplingPortRatePolicy ActionEnum	0..1	attr	Defines the action to be performed when this rate policy is violated. **Tags:** atp.Status=obsolete
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. **Tags:** atp.Status=obsolete
timeInterval	TimeValue	0..1	attr	Time interval used to define the base of the rate policy. **Tags:** atp.Status=obsolete
vLan	EthernetPhysical Channel	*	ref	Defines the VLANs this rate policy shall be limited on. If no VLAN is given this rate policy is not considering VLAN tags. **Tags:** atp.Splitkey=vLan atp.Status=obsolete

Table A.243: CouplingPortRatePolicy

Enumeration	**CouplingPortRoleEnum**
Note	Defines the role a CouplingPort takes in the context of a CouplingElement.
Aggregated by	CouplingPort.couplingPortRole
Literal	***Description***
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 CoupingPort 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 A.244: CouplingPortRoleEnum

Class	**CouplingPortScheduler**			
Note	Defines a scheduler for the CouplingPort egress structure.			
Base	*ARObject*, *CouplingPortStructuralElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortDetails.couplingPortStructuralElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**CouplingPortScheduler**			
portScheduler	EthernetCouplingPort SchedulerEnum	0..1	attr	Defines the schedule algorithm to be used.
predecessor (ordered)	CouplingPortStructural Element	*	ref	Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.

Table A.245: CouplingPortScheduler

Class	**CouplingPortTrafficClassAssignment**			
Note	Defines the assignment of Traffic Class to a frame.			
Base	*ARObject*, *Referrable*			
Aggregated by	CouplingPortDetails.ethernetTrafficClassAssignment			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
priority	PositiveInteger	*	attr	Defines a priority which is mapped onto a Traffic Class.
trafficClass	PositiveInteger	0..1	attr	Defines the Traffic Class which is assigned.

Table A.246: CouplingPortTrafficClassAssignment

Class	**CpSoftwareCluster**			
Note	This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings. **Tags:** atp.recommendedPackage=CpSoftwareClusters This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster.
swComponent Assignment	SwComponent PrototypeAssignment	*	aggr	This is the collection of SwComponentPrototype Assignments **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentAssignment, swComponent Assignment.variationPoint.shortLabel vh.latestBindingTime=postBuild
swComposition	CompositionSw ComponentType	*	ref	Software Components in the context of a CompositionSw ComponentType belonging to this CP Software Cluster. This reference can be used to describe the belonging SWCs when the CP Software Cluster is described out of the context of a System, e.g. reusable CP Software Cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComposition, swComposition.composition SwComponentType, swComposition.variationPoint.short Label vh.latestBindingTime=systemDesignTime

Table A.247: CpSoftwareCluster

Class	CpSoftwareClusterBinaryManifestDescriptor			
Note	This meta-class has the ability to act as a hub for all information related to the binary manifest of a given CP software cluster. The manifest is subject to integrator work and therefore not a part of the definition of the CP software cluster itself. **Tags:** atp.recommendedPackage=CpSoftwareClusterBinaryManifestDescriptors This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cpSoftware Cluster	CpSoftwareCluster	0..1	ref	This reference identifies the CpSoftwareCluster to which the enclosing CpSoftwareClusterBinaryManifest Descriptor belongs, The CpSoftwareClusterBinaryManifestDescriptor is defined in an integration phase while the referenced Cp SoftwareCluster represents a design element. Therefore, it makes sense to use a reference rather than an aggregation in the relation of the two meta-classes. **Tags:** atp.Splitkey=cpSoftwareCluster
metaDataField	BinaryManifestMeta DataField	*	aggr	This aggregation identifies the collection of meta-data contained in the enclosing binary manifest. **Tags:** atp.Splitkey=metaDataField, metaDataField.short Name
provide Resource	BinaryManifestProvide Resource	*	aggr	This aggregation represents the collection of provided resources in the enclosing binary manifest. **Tags:** atp.Splitkey=provideResource, provide Resource.shortName
require Resource	BinaryManifestRequire Resource	*	aggr	This aggregation represents the collection of required resources in the enclosing binary manifest. **Tags:** atp.Splitkey=requireResource, require Resource.shortName
resource Definition	BinaryManifest ResourceDefinition	*	aggr	This aggregation represents the collection of binary manifest resource definitions that belong to the enclosing CpSoftwareClusterBinaryManifestDescriptor. **Tags:** atp.Splitkey=resourceDefinition, resource Definition.shortName
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster. This id is assigned by an integrator, but may also be copied from CpSoftware Cluster.softwareClusterId if available.

Table A.248: CpSoftwareClusterBinaryManifestDescriptor

Class	CpSoftwareClusterCommunicationResource			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**CpSoftwareClusterCommunicationResource**			
communication ResourceProps	CpSoftwareCluster Communication ResourceProps	0..1	aggr	This aggregation supports the further qualification of the enclosing CpSoftwareClusterCommunicationRecource by means of additional attributes depending on the nature of the CpSoftwareClusterCommunicationRecource. **Tags:** atp.Splitkey=communicationResourceProps

Table A.249: CpSoftwareClusterCommunicationResource

Class	***CpSoftwareClusterResource*** (abstract)			
Note	Represents a single resource required or provided by a CP Software Cluster. **Tags:** atp.recommendedPackage=Resources This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CpSoftwareClusterCommunicationResource, CpSoftwareClusterServiceResource			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dependent Resource	RoleBasedResource Dependency	*	aggr	Link to a resource which depends on this resource to implement them. **Tags:** atp.Splitkey=dependentResource
globalResource Id	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines.
isMandatory	Boolean	0..1	attr	This attribute indicates, that the resource is mandatory to operate the Software Cluster. If the resource is not provided on the machine the connection process of any Software Cluster requiring this resource gets aborted.

Table A.250: CpSoftwareClusterResource

Class	**CpSoftwareClusterResourceToApplicationPartitionMapping**			
Note	This meta class maps a Software Cluster resource to an Application Partition to restrict the usage. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.resourceToApplicationPartitionMapping, SystemMapping.resourceTo ApplicationPartitionMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Partition	ApplicationPartition	0..1	ref	ApplicationPartition for which the mapping applies. **Tags:** atp.Splitkey=applicationPartition
resource	CpSoftwareCluster Resource	0..1	ref	Software Cluster Resource for which the mapping applies. **Tags:** atp.Splitkey=resource

Table A.251: CpSoftwareClusterResourceToApplicationPartitionMapping

Class	CpSoftwareClusterServiceResource			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the BSW. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
resourceNeeds	EcucContainerValue	*	ref	Refernce(s) to one or multiple EcucContainerValue(s) qualifying the characteristics of the resource. **Tags:** atp.Splitkey=resourceNeeds

Table A.252: CpSoftwareClusterServiceResource

Class	CpSoftwareClusterToApplicationPartitionMapping			
Note	This meta class defines ApplicationPartitions that are applicable for the CpSoftwareCluster. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.softwareClusterToApplicationPartitionMapping, SystemMapping.softwareClusterToApplicationPartitionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Partition	ApplicationPartition	*	ref	Collection of ApplicationPartitions available in the Cp SoftwareCluster **Tags:** atp.Splitkey=applicationPartition
softwareCluster	CpSoftwareCluster	0..1	ref	Software Cluster Resource for which the mapping applies **Tags:** atp.Splitkey=softwareCluster

Table A.253: CpSoftwareClusterToApplicationPartitionMapping

Class	CpSoftwareClusterToEcuInstanceMapping			
Note	This meta class maps a CpSoftwareCluster to a EcuInstance. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swClusterMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecuInstance	EcuInstance	0..1	ref	Reference to a specific ECU Instance description. **Tags:** atp.Splitkey=ecuInstance
machineId	PositiveInteger	0..1	attr	Unique number of the (virtual or physical) machine to which the Software Cluster is mapped.
swCluster	CpSoftwareCluster	*	ref	The mapped CP Software Cluster **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swCluster, swCluster.cpSoftwareCluster, swCluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table A.254: CpSoftwareClusterToEcuInstanceMapping

Class	CpSoftwareClusterToResourceMapping			
Note	This meta class maps a service resource to CP Software Clusters. By this mapping it's specified whether the Software Cluster has to provide or to require the resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.softwareClusterToResourceMapping, SystemMapping.softwareClusterToResourceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provider	CpSoftwareCluster	0..1	ref	CP Software Cluster providing the resource **Tags:** atp.Splitkey=provider
requester	CpSoftwareCluster	*	ref	CP Software Cluster requesting the resource **Tags:** atp.Splitkey=requester
service Resource	CpSoftwareCluster ServiceResource	0..1	ref	Service resource for which the mapping applies. **Tags:** atp.Splitkey=serviceResource

Table A.255: CpSoftwareClusterToResourceMapping

Class	CryptoServiceCertificate			
Note	This meta-class represents the ability to model a cryptographic certificate. **Tags:** atp.recommendedPackage=CryptoServiceCertificates			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
algorithmFamily	CryptoCertificate AlgorithmFamilyEnum	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	CryptoCertificateFormat Enum	0..1	attr	This attribute can be used to provide information about the format used to create the certificate
maximum Length	PositiveInteger	0..1	attr	This attribute represents the ability to define the maximum length of the certificate in bytes.
nextHigher Certificate	CryptoService Certificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain. **Tags:** atp.Splitkey=nextHigherCertificate
serverName Identification	String	0..1	attr	Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.

Table A.256: CryptoServiceCertificate

Class	CryptoServiceKey			
Note	This meta-class has the ability to represent a crypto key. **Tags:** atp.recommendedPackage=CryptoDevelopmentKeys			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**CryptoServiceKey**			
algorithmFamily	String	0..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. **Tags:** atp.Splitkey=developmentValue
keyGeneration	CryptoServiceKey GenerationEnum	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	0..1	attr	This attribute describes the length of the cryptographic key in bits.

Table A.257: CryptoServiceKey

Class	**CryptoServicePrimitive**			
Note	This meta-class has the ability to represent a crypto primitive. **Tags:** atp.recommendedPackage=CryptoPrimitives			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
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.
algorithm Secondary Family	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 A.258: CryptoServicePrimitive

Class	**CryptoServiceQueue**			
Note	This meta-class has the ability to represent a crypto queue. **Tags:** atp.recommendedPackage=CryptoServiceQueues			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
queueSize	PositiveInteger	0..1	attr	Defines the queue size of the CryptoServiceQueue.

Table A.259: CryptoServiceQueue

Class	**CycleCounter**			
Note	The communication cycle where the frame is send is described by the attribute "cycleCounter".			
Base	*ARObject*, *CommunicationCycle*			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
CycleCounter	Integer	0..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 A.260: CycleCounter

Class	**CycleRepetition**			
Note	The communication cycle where the frame is send is described by the attributes baseCycle and cycle Repetition.			
Base	*ARObject*, *CommunicationCycle*			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
BaseCycle	Integer	0..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	0..1	attr	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.

Table A.261: CycleRepetition

Class	**CyclicHandlingComDataToOsTaskProxyMapping**			
Note	This meta-class is used to map VariableDataPrototypes to an OsTaskProxy for the Cyclic Handling of Communication Data in the RTE. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cyclicHandlingComDataToOsTaskProxyMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
e2eNoNewData ExtensionFactor	PositiveInteger	0..1	attr	This attribute defines the jitter factor of E2E no new data reception cycle time.
e2e Oversampling Period	TimeValue	0..1	attr	This attribute defines an optional E2E oversampling cycle time for the cyclic processing of the mapped VariableData Prototypes.
offset	PositiveInteger	0..1	attr	This attribute is used to describe the position in the Os Task as a relative value.
osTaskProxy	OsTaskProxy	0..1	ref	Reference to the OsTaskProxy. **Tags:** atp.Splitkey=osTaskProxy
rxCycleProcess Time	TimeValue	0..1	attr	This attribute defines the cyclic processing time within the OsTask.

▽

△

Class	CyclicHandlingComDataToOsTaskProxyMapping			
variableData Prototype	VariableDataPrototype	*	iref	Reference to the VariableDataPrototypes for which the "Cyclic Handling of Communication Data in the RTE" will be implemented by the RTE. **Tags:** atp.Splitkey=variableDataPrototype, variableData Prototype.contextComponent, variableData Prototype.contextComposition, variableData Prototype.contextPort, variableDataPrototype.targetData Prototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef

Table A.262: CyclicHandlingComDataToOsTaskProxyMapping

Class	CyclicTiming			
Note	Specification of a cyclic sending behavior.			
Base	*ARObject*, *Describable*			
Aggregated by	TransmissionModeTiming.cyclicTiming			
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. **Tags:** atp.Splitkey=timeOffset
timePeriod	TimeRangeType	0..1	aggr	Period of the repetition of cyclic transmissions. **Tags:** atp.Splitkey=timePeriod

Table A.263: CyclicTiming

Class	DataConstr			
Note	This meta-class represents the ability to specify constraints on data. **Tags:** atp.recommendedPackage=DataConstrs			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. **Tags:** atp.Splitkey=dataConstrRule xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.264: DataConstr

Class	DataConstrRule
Note	This meta-class represents the ability to express one specific data constraint rule.
Base	*ARObject*
Aggregated by	DataConstr.dataConstrRule

▽

△

Class	**DataConstrRule**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
constrLevel	Integer	0..1	attr	This attribute describes the category of a constraint. One of its functions is in the area of constraint violation, where it can be used from a certain level, to produce error messages. The lower the level, the more stringent the check. Used to distinguish hard or soft limits. **Tags:** xml.sequenceOffset=20
internalConstrs	InternalConstrs	0..1	aggr	Describes the limitations applicable on the internal domain (as opposed to the physical domain). **Tags:** atp.Splitkey=internalConstrs xml.sequenceOffset=40
physConstrs	PhysConstrs	0..1	aggr	Describes the limitations applicable on the physical domain (as opposed to the internal domain). **Tags:** atp.Splitkey=physConstrs xml.sequenceOffset=30

Table A.265: DataConstrRule

Class	**DataDumpEntry**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
byteValue (ordered)	Integer	*	attr	Supplier specific format.

Table A.266: DataDumpEntry

Class	**DataFilter**			
Note	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.			
Base	*ARObject*			
Aggregated by	ISignalPort.dataFilter, NonqueuedReceiverComSpec.filter, NonqueuedSenderComSpec.dataFilter, SignalBasedEventElementToISignalTriggeringMapping.filter, SignalBasedFieldToISignalTriggering Mapping.filter, SignalServiceTranslationElementProps.filter, TransmissionModeCondition.dataFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilterType	DataFilterTypeEnum	0..1	attr	This attribute specifies the type of the filter.
mask	UnlimitedInteger	0..1	attr	Mask for old and new value.
max	UnlimitedInteger	0..1	attr	Value to specify the upper boundary
min	UnlimitedInteger	0..1	attr	Value to specify the lower boundary
offset	PositiveInteger	0..1	attr	Specifies the initial number of messages to occur before the first message is passed
period	PositiveInteger	0..1	attr	Specifies number of messages to occur before the message is passed again
x	UnlimitedInteger	0..1	attr	Value to compare with

Table A.267: DataFilter

Enumeration	**DataFilterTypeEnum**
Note	This enum specifies the supported DataFilterTypes.
Aggregated by	DataFilter.dataFilterType
Literal	***Description***
always	No filtering is performed so that the message always passes. **Tags:** atp.EnumerationLiteralIndex=0
maskedNewDiffers MaskedOld	Pass messages where the masked value has changed. (new_value&mask) !=(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	Pass messages whose masked value is not equal to a specific value x (new_value&mask) != x new_value: current value of the message **Tags:** atp.EnumerationLiteralIndex=2
maskedNewEquals X	Pass messages whose masked value is equal to a specific value x (new_value&mask) == x new_value: current value of the message **Tags:** atp.EnumerationLiteralIndex=3
never	The filter removes all messages. **Tags:** atp.EnumerationLiteralIndex=4
newIsOutside	Pass a message if its value is outside a predefined boundary. (min > new_value) OR (new_value > max) **Tags:** atp.EnumerationLiteralIndex=5
newIsWithin	Pass a message if its value is within a predefined boundary. min <= new_value <= max **Tags:** atp.EnumerationLiteralIndex=6
oneEveryN	Pass a message once every N message occurrences. Algorithm: occurrence % period == offset Start: occurrence = 0. 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

Table A.268: DataFilterTypeEnum

Enumeration	**DataIdModeEnum**
Note	Supported inclusion modes to include the implicit two-byte Data ID in the one-byte CRC.
Aggregated by	E2EProfileConfiguration.dataIdMode, EndToEndTransformationDescription.dataIdMode
Literal	***Description***
all16Bit	Two bytes are included in the CRC (double ID configuration). **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**DataIdModeEnum**
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 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
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 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.

Table A.269: DataIdModeEnum

Class	***DataInterface*** (abstract)			
Note	The purpose of this meta-class is to act as an abstract base class for subclasses that share the semantics of being concerned about data (as opposed to e.g. operations).			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Subclasses	NvDataInterface, ParameterInterface, SenderReceiverInterface			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.270: DataInterface

Class	***DataMapping*** (abstract)			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	*ARObject*			
Subclasses	ClientServerToSignalMapping, SenderReceiverToSignalGroupMapping, SenderReceiverToSignal Mapping, TriggerToSignalMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping. **Tags:** atp.Splitkey=introduction

Table A.271: DataMapping

Class	***DataPrototype*** (abstract)			
Note	Base class for prototypical roles of any data type.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*ApplicationCompositeElementDataPrototype*, *AutosarDataPrototype*			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***DataPrototype*** (abstract)			
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level. **Tags:** atp.Splitkey=swDataDefProps

Table A.272: DataPrototype

Class	**DataPrototypeInClientServerInterfaceInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*, *DataPrototypeInPortInterfaceInstanceRef*			
Aggregated by	DataPrototypeInPortInterfaceRef.dataPrototypeInClientServerInterface, DiagnosticServiceSw Mapping.accessedDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	ClientServerInterface	0..1	ref	**Stereotypes:** atpDerived
contextData PrototypeInCs (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=20
rootData PrototypeInCs	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=10
targetData PrototypeInCs	DataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=30

Table A.273: DataPrototypeInClientServerInterfaceInstanceRef

Class	**DataPrototypeInPortInterfaceRef**			
Note	This class represents a RootDataPrototype that is typed by an ApplicationDataType or Implementation DataType or a DataTypeElement that is aggregated within a composite application data type (record or array).			
Base	*ARObject*, *DataPrototypeReference*			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataPrototypeIn ClientServer Interface	DataPrototype	0..1	iref	This element defines a reference to a DataPrototype in the context of a ClientServerInterface. **Tags:** atp.Splitkey=dataPrototypeInClientServerInterface, dataPrototypeInClientServerInterface.rootDataPrototype InCs, dataPrototypeInClientServerInterface.contextData PrototypeInCs, dataPrototypeInClientServer Interface.targetDataPrototypeInCs **InstanceRef implemented by:** DataPrototypeInClient ServerInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**DataPrototypeInPortInterfaceRef**			
dataPrototypeIn SenderReceiver Interface	DataPrototype	0..1	iref	This element defines a reference to a DataPrototype in the context of a SenderReceiverInterface. **Tags:** atp.Splitkey=dataPrototypeInSenderReceiver Interface, dataPrototypeInSenderReceiverInterface.root DataPrototypeInSr, dataPrototypeInSenderReceiver Interface.contextDataPrototypeInSr, dataPrototypeIn SenderReceiverInterface.targetDataPrototypeInSr **InstanceRef implemented by:** DataPrototypeInSender ReceiverInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table A.274: DataPrototypeInPortInterfaceRef

Class	**DataPrototypeInSenderReceiverInterfaceInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*, *DataPrototypeInPortInterfaceInstanceRef*			
Aggregated by	DataPrototypeInPortInterfaceRef.dataPrototypeInSenderReceiverInterface			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
base	SenderReceiver Interface	0..1	ref	**Stereotypes:** atpDerived
contextData PrototypeInSr (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=20
rootData PrototypeInSr	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=10
targetData PrototypeInSr	DataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=30

Table A.275: DataPrototypeInSenderReceiverInterfaceInstanceRef

Class	**DataPrototypeMapping**			
Note	Defines the mapping of two particular `VariableDataPrototype`s, `ParameterDataPrototype`s or `ArgumentDataPrototype`s with non-equal `shortName`s, non-equal structure (specific condition is described by [constr_1187]), and/or non-equal semantic (resolution or range) in context of two different `SenderReceiverInterface`, `NvDataInterface` or `ParameterInterface` or Operations. If the semantic is unequal, the following rules apply: The `textTableMapping` is only applicable if the referred `DataPrototype`s are typed by `AutosarDataType` referring to `CompuMethod`s of `category` `TEXTTABLE`, `SCALE_LINEAR_AND_TEXTTABLE` or `BITFIELD_TEXTTABLE`. In the case that the `DataPrototype`s are typed by `AutosarDataType` either referring to `CompuMethod`s 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 `Unit`s and the `CompuRationalCoeffs` of a `compuInternalToPhys` of the referred `CompuMethod`s.			
Base	*ARObject*			
Aggregated by	ClientServerOperationMapping.argumentMapping, VariableAndParameterInterfaceMapping.dataMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
firstData Prototype	AutosarDataPrototype	0..1	ref	First to be mapped DataPrototype in context of a Sender ReceiverInterface, NvDataInterface, ParameterInterface or Operation. **Tags:** atp.Splitkey=firstDataPrototype

▽

△

Class	**DataPrototypeMapping**			
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. This reference also specifies the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain (i.e. from the DataPrototype Mapping.secondDataPrototype to the DataPrototype Mapping.firstDataPrototype) if the referenced Data Transformation is symmetric, i.e. attribute Data Transformation.dataTransformationKind is set to symmetric. **Tags:** atp.Splitkey=firstToSecondDataTransformation
secondData Prototype	AutosarDataPrototype	0..1	ref	Second to be mapped DataPrototype in context of a SenderReceiverInterface, NvDataInterface, Parameter Interface or Operation. **Tags:** atp.Splitkey=secondDataPrototype
secondToFirst Data Transformation	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 Data PrototypeMapping.firstDataPrototype. **Tags:** atp.Splitkey=secondToFirstDataTransformation
subElement Mapping	SubElementMapping	*	aggr	This represents the owned SubelementMapping. **Tags:** atp.Splitkey=subElementMapping
textTable Mapping	TextTableMapping	*	aggr	Applied TextTableMapping(s) **Tags:** atp.Splitkey=textTableMapping

Table A.276: DataPrototypeMapping

Class	***DataPrototypeReference*** (abstract)			
Note	This meta-class provides the ability to reference a DataPrototype.			
Base	*ARObject*			
Subclasses	DataPrototypeInPortInterfaceRef, ImplementationDataTypeElementInPortInterfaceRef			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.277: DataPrototypeReference

Class	**DataPrototypeTransformationProps**			
Note	DataPrototypeTransformationProps allows to set the attributes for the different Transformation Technologies that are DataPrototype specific.			
Base	*ARObject*			
Aggregated by	*TransformationISignalProps*.dataPrototypeTransformationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	DataPrototypeTransformationProps			
dataPrototypeIn PortInterface Ref	DataPrototype Reference	0..1	aggr	Reference to a DataPrototype that is transported in the serialized ISignal. **Tags:** atp.Splitkey=dataPrototypeInPortInterfaceRef
ident	DataPrototype TransformationProps Ident	0..1	aggr	This adds the ability to add a shortName to DataPrototype TransformationProps. Please note that the short-name needs to be provided if the splitable mechanism is used. **Stereotypes:** atpIdentityContributor
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. **Tags:** atp.Splitkey=networkRepresentationProps
transformation Props	TransformationProps	0..1	ref	Collection of AutosarDataPrototype related configuration settings for a transformer. **Tags:** atp.Splitkey=transformationProps

Table A.278: DataPrototypeTransformationProps

Class	DataReceiveErrorEvent			
Note	This event is raised when the Com layer detects and notifies an error concerning the reception of the referenced `VariableDataPrototype`.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data	VariableDataPrototype	0..1	iref	The referenced VariableDataPrototype raises this DataReceiveErrorEvent when there was an error during the reception. **Tags:** atp.Splitkey=data, data.contextRPort, data.targetDataElement **InstanceRef implemented by:** RVariableInAtomicSwcInstanceRef

Table A.279: DataReceiveErrorEvent

Class	DataReceivedEvent			
Note	This event is raised when the referenced data element is received.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data	VariableDataPrototype	0..1	iref	The referenced VariableDataPrototype raises this DataReceivedEvent when the data has been received. **Tags:** atp.Splitkey=data, data.contextRPort, data.targetDataElement **InstanceRef implemented by:** RVariableInAtomicSwcInstanceRef

Table A.280: DataReceivedEvent

Class	DataSendCompletedEvent			
Note	This event is raised when the referenced explicit data element has been sent or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	VariableAccess	0..1	ref	The referenced `VariableAccess` raises this `DataSendCompletedEvent` when the explicit write access was successful or an error occurred. **Tags:** atp.Splitkey=eventSource

Table A.281: DataSendCompletedEvent

Class	DataTransformation			
Note	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.dataTransformation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
data Transformation Kind	DataTransformationKind Enum	0..1	attr	This attribute controls the kind of DataTransformation to be applied.
executeDespite Data Unavailability	Boolean	0..1	attr	Specifies whether the transformer chain is executed even if no input data are available.
transformer Chain (ordered)	Transformation Technology	*	ref	This attribute represents the definition of a chain of transformers that are supposed to be executed according to the order of being referenced from DataTransformation.

Table A.282: DataTransformation

Enumeration	DataTransformationKindEnum
Note	This enumeration contributes to the definition of the scope of the DataTransformation.
Aggregated by	DataTransformation.dataTransformationKind
Literal	**Description**
asymmetricFrom ByteArray	The DataTransformation shall only be applied to the receiving end only, i.e. transform from byte array to data type. **Tags:** atp.EnumerationLiteralIndex=0
asymmetricToByte Array	The DataTransformation shall be applied to the sending end only, i.e. from data type to byte array. **Tags:** atp.EnumerationLiteralIndex=1
symmetric	The DataTransformation shall be applied at both the sending and the receiving end of the communication. **Tags:** atp.EnumerationLiteralIndex=2

Table A.283: DataTransformationKindEnum

Class	DataTypeMap			
Note	This class represents the relationship between `ApplicationDataType` and its implementing `AbstractImplementationDataType`.			
Base	*ARObject*			
Aggregated by	DataTypeMappingSet.dataTypeMap			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**DataTypeMap**			
applicationDataType	ApplicationDataType	0..1	ref	This is the corresponding `ApplicationDataType` **Tags:** atp.Splitkey=applicationDataType
implementationDataType	AbstractImplementationDataType	0..1	ref	This is the corresponding `AbstractImplementationDataType`. **Tags:** atp.Splitkey=implementationDataType

Table A.284: DataTypeMap

Class	**DataTypeMappingSet**			
Note	This class represents a list of mappings between `ApplicationDataType`s and `ImplementationDataType`s. In addition, it can contain mappings between `ImplementationDataType`s and `ModeDeclarationGroup`s. **Tags:** atp.recommendedPackage=DataTypeMappingSets			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an `ApplicationDataType` and its `AbstractImplementationDataType`. **Tags:** atp.Splitkey=dataTypeMap
modeRequestTypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an `ModeDeclarationGroup` and its `AbstractImplementationDataType`. **Tags:** atp.Splitkey=modeRequestTypeMap

Table A.285: DataTypeMappingSet

Enumeration	**DataTypePolicyEnum**
Note	This class lists the supported DataTypePolicies.
Aggregated by	ISignal.dataTypePolicy
Literal	***Description***
ddsService	This literal indicates that this ISignal is used to transport a message as part of a service for Dds. **Tags:** atp.EnumerationLiteralIndex=6 atp.Status=candidate
ddsSignal	This literal indicates that this ISignal is used to transport a signal based signal for Dds. **Tags:** atp.EnumerationLiteralIndex=5 atp.Status=candidate
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

▽

△

Enumeration	**DataTypePolicyEnum**
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 (networkRepresentationFromCom Spec) are not fullfilled 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 A.286: DataTypePolicyEnum

Class	**DataWriteCompletedEvent**			
Note	This event is raised when an implicit write access was successful or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
eventSource	VariableAccess	0..1	ref	The referenced `VariableAccess` raises this `DataWriteCompletedEvent` when the implicit write access was successful or an error occurred. **Tags:** atp.Splitkey=eventSource

Table A.287: DataWriteCompletedEvent

Class	**DcmIPdu**			
Note	Represents the IPdus handled by Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
diagPduType	DiagPduType	0..1	attr	Attribute is used to distinguish a request from a response.

Table A.288: DcmIPdu

Class	**DdsCpConsumedServiceInstance**			
Note	This meta-class represents the ability to describe the existence and configuration of a consumed (required) service instance in a concrete implementation on top of DDS. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *DdsCpServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**DdsCpConsumedServiceInstance**			
consumedEvent	DdsServiceInstance EventCp	*	aggr	Collection of consumed events. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedEvent, consumedEvent.shortName, consumedEvent.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
consumedField	DdsServiceInstance FieldCp	*	aggr	Collection of consumed fields. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedField, consumedField.shortName, consumedField.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
consumed Operation	DdsServiceInstance OperationCp	*	aggr	Collection of consumed operations. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedOperation, consumedOperation.shortName, consumedOperation.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
localUnicast Address	ApplicationEndpoint	0..1	ref	The local address over which the Service is consumed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicastAddress.applicationEndpoint, localUnicastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.namePlural=LOCAL-UNICAST-ADDRESSES
requiredService InstanceId	PositiveInteger	0..1	attr	This attribute describes the required service instance ID. **Tags:** atp.Status=candidate
staticRemote Multicast Address	ApplicationEndpoint	0..1	ref	This reference defines the remote multicast address of the Service provider. This reference shall ONLY be used if the remote multicast address of the server is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteMulticastAddress, staticRemoteMulticastAddress.applicationEndpoint, staticRemoteMulticastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-MULTICAST-ADDRESSES

▽

△

Class	**DdsCpConsumedServiceInstance**			
staticRemote UnicastAddress	ApplicationEndpoint	0..1	ref	This reference defines the remote unicast address of the Service provider. This reference shall ONLY be used if the remote unicast address of the server is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteUnicastAddress, staticRemote UnicastAddress.applicationEndpoint, staticRemote UnicastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-UNICAST-ADDRESSES

Table A.289: DdsCpConsumedServiceInstance

Class	**DdsCpDomain**			
Note	Definition of a DDS Domain. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConfig.ddsDomain			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ddsPartition	DdsCpPartition	*	aggr	Collection of DDS Partition definitions. **Tags:** atp.Splitkey=ddsPartition, ddsPartition.shortName atp.Status=candidate
ddsTopic	DdsCpTopic	*	aggr	Collection of DDS Topics. **Tags:** atp.Splitkey=ddsTopic, ddsTopic.shortName atp.Status=candidate
domainId	PositiveInteger	0..1	attr	Definition of the DDS Domain Id. **Tags:** atp.Status=candidate

Table A.290: DdsCpDomain

Class	**DdsCpISignalToDdsTopicMapping**			
Note	Mapping of an ISignal to a DdsTopic. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	SystemMapping.ddsISignalToTopicMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ddsTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic. **Tags:** atp.Splitkey=ddsTopic atp.Status=candidate

▽

△

Class	**DdsCpISignalToDdsTopicMapping**			
iSignal	ISignal	0..1	ref	Reference to the ISignal. **Tags:** atp.Splitkey=iSignal atp.Status=candidate

Table A.291: DdsCpISignalToDdsTopicMapping

Class	**DdsCpPartition**			
Note	Definition of a DDS Partition. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpDomain.ddsPartition			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
partitionName	String	0..1	attr	Definition of the DDS Partition Name. '*' may be used to define the default partition. **Tags:** atp.Status=candidate

Table A.292: DdsCpPartition

Class	**DdsCpProvidedServiceInstance**			
Note	This meta-class represents the ability to describe the existence and configuration of a provided service instance in a concrete implementation on top of DDS. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *DdsCpServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
localUnicast Address	ApplicationEndpoint	0..1	ref	The local address over which the Service is provided. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.namePlural=LOCAL-UNICAST-ADDRESSES
providedEvent	DdsServiceInstance EventCp	*	aggr	Collection of provided events. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedEvent, providedEvent.shortName, providedEvent.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
providedField	DdsServiceInstance FieldCp	*	aggr	Collection of provided fields. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedField, providedField.shortName, providedField.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild

▽

△

Class	**DdsCpProvidedServiceInstance**			
provided Operation	DdsServiceInstance OperationCp	*	aggr	Collection of provided operations. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedOperation, providedOperation.shortName, providedOperation.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
providedServiceInstanceId	PositiveInteger	0..1	attr	This attribute describes the provided service instance ID. **Tags:** atp.Status=candidate
staticRemoteMulticastAddress	ApplicationEndpoint	0..1	ref	This reference defines the remote multicast address of Service consumers. This reference shall ONLY be used if the remote multicast address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteMulticastAddress, staticRemoteMulticastAddress.applicationEndpoint, staticRemoteMulticastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-MULTICAST-ADDRESSES
staticRemoteUnicastAddress	ApplicationEndpoint	*	ref	This reference defines the remote unicast addresses of Service consumers. This reference shall ONLY be used if the remote unicast address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteUnicastAddress, staticRemoteUnicastAddress.applicationEndpoint, staticRemoteUnicastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-UNICAST-ADDRESSES

Table A.293: DdsCpProvidedServiceInstance

Class	***DdsCpServiceInstance*** (abstract)			
Note	Provided and Consumed Dds Service Instances that are available at the ApplicationEndpoint. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DdsCpConsumedServiceInstance, DdsCpProvidedServiceInstance			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsFieldReplyTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of field setters. **Tags:** atp.Splitkey=ddsFieldReplyTopic atp.Status=candidate

▽

△

Class	***DdsCpServiceInstance*** (abstract)			
ddsField RequestTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of field getters. **Tags:** atp.Splitkey=ddsFieldRequestTopic atp.Status=candidate
ddsMethod ReplyTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of method replies. **Tags:** atp.Splitkey=ddsMethodReplyTopic atp.Status=candidate
ddsMethod RequestTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of method requests. **Tags:** atp.Splitkey=ddsMethodRequestTopic atp.Status=candidate
ddsServiceQos Profile	DdsCpQosProfile	0..1	ref	Reference to the QOS Profile used for the service. **Tags:** atp.Splitkey=ddsServiceQosProfile atp.Status=candidate
discoveryType	DdsDiscoveryType Enum	0..1	attr	Defines the discovery scheme for this DDS Service Instance.
resource IdentifierType	DdsResourceIdentifier TypeEnum	0..1	attr	Defines the in-band instance identification fields used to discriminate samples related to specific Service Instances sharing the same DDS Topics. **Tags:** atp.Status=candidate
serviceInterface Id	String	0..1	attr	Unique Identifier that identifies the ServiceInterface in DDS. This Identifier is encoded in the USER_DATA QoS of the DomainParticipant associated with the Service Instance and its value is propagated by DDS Discovery messages. **Tags:** atp.Status=candidate
serviceMinor Version	PositiveInteger	0..1	attr	Minor Version of the Service defined by this DdsCp ServiceInstance. **Tags:** atp.Status=candidate
transport Protocol	String	0..1	attr	This attribute defines which Transport Layer Protocol(s) this Instance is intended to use. **Tags:** atp.Status=candidate

Table A.294: DdsCpServiceInstance

Class	**DdsCpTopic**			
Note	Definition of a DDS Topic. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpDomain.ddsTopic			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsPartition	DdsCpPartition	0..1	ref	Reference to the DDS Partition this topic is communicated. **Tags:** atp.Splitkey=ddsPartition atp.Status=candidate
topicName	String	0..1	attr	Definition of the DDS Topic Name. **Tags:** atp.Status=candidate

Table A.295: DdsCpTopic

Class	**DdsServiceInstanceEventCp**			
Note	This element represents an event as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedEvent, DdsCpProvidedServiceInstance.providedEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsEventQos Profile	DdsCpQosProfile	0..1	ref	Reference to the QOS Profile used for this Event. **Tags:** atp.Splitkey=ddsEventQosProfile atp.Status=candidate
ddsEventTopic	DdsCpTopic	0..1	ref	Reference to the DDS Topic used for this Event. **Tags:** atp.Splitkey=ddsEventTopic atp.Status=candidate
eventTriggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsEvent message. **Tags:** atp.Splitkey=eventTriggering atp.Status=candidate

Table A.296: DdsServiceInstanceEventCp

Class	**DdsServiceInstanceFieldCp**			
Note	This element represents a field as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedField, DdsCpProvidedServiceInstance.providedField			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
getter	DdsServiceInstance OperationCp	0..1	ref	Reference to the field getter operation. **Tags:** atp.Splitkey=getter atp.Status=candidate

▽

△

Class	**DdsServiceInstanceFieldCp**			
notifier	DdsServiceInstance EventCp	0..1	ref	Reference to the field notifier event. **Tags:** atp.Splitkey=notifier atp.Status=candidate
setter	DdsServiceInstance OperationCp	0..1	ref	Reference to the field setter operation. **Tags:** atp.Splitkey=setter atp.Status=candidate

Table A.297: DdsServiceInstanceFieldCp

Class	**DdsServiceInstanceOperationCp**			
Note	This element represents an operation as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedOperation, DdsCpProvidedServiceInstance.provided Operation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation Request Triggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsOperation request message. **Tags:** atp.Splitkey=operationRequestTriggering atp.Status=candidate
operation Response Triggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsOperation response message. **Tags:** atp.Splitkey=operationResponseTriggering atp.Status=candidate

Table A.298: DdsServiceInstanceOperationCp

Class	**DdsTransformationDescription**			
Note	The DdsTransformationDescription is used to specify the DDS transformer specific attributes. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.299: DdsTransformationDescription

Class	DefaultValueElement			
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	*ARObject*			
Aggregated by	PduMappingDefaultValue.defaultValueElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
elementByte Value	Integer	0..1	attr	The integer value of a freely defined data byte.
elementPosition	Integer	0..1	attr	This attribute specifies the byte position of the element within the default value

Table A.300: DefaultValueElement

Class	DelegatedPortAnnotation			
Note	Annotation to a "delegated port" to specify the Signal Fan In or Signal Fan Out inside the CompositionSw ComponentType.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.delegatedPortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
signalFan	SignalFanEnum	0..1	attr	Specifies the Signal Fan In or Signal Fan Out inside the Composition Type.

Table A.301: DelegatedPortAnnotation

Class	DelegationSwConnector			
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*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
innerPort	PortPrototype	0..1	iref	The port that belongs to the ComponentPrototype in the composition **Tags:** atp.Splitkey=innerPort, innerPort.context Component, innerPort.targetPPort, innerPort.targetRPort **InstanceRef implemented by:** PortInCompositionType InstanceRef
outerPort	PortPrototype	0..1	ref	The port that is located on the outside of the Composition Type **Tags:** atp.Splitkey=outerPort

Table A.302: DelegationSwConnector

Class	DependencyOnArtifact			
Note	Dependency on the existence of another artifact, e.g. a library.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Implementation*.generatedArtifact, *Implementation*.requiredArtifact, *Implementation*.requiredGeneratorTool			
Attribute	**Type**	**Mult.**	**Kind**	**Note**



△

Class	**DependencyOnArtifact**			
artifact Descriptor	AutosarEngineering Object	0..1	aggr	The specified artifact needs to exist. **Tags:** atp.Splitkey=artifactDescriptor
usage	DependencyUsage Enum	*	attr	Specification for which process step(s) this dependency is required.

Table A.303: DependencyOnArtifact

Class	**Dhcpv6Props**			
Note	This meta-class specifies the configuration options for DHCPv6.			
Base	*ARObject*			
Aggregated by	Ipv6Props.dhcpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpDhcpV6 CnfDelayMax	TimeValue	0..1	attr	Maximum delay in seconds 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.
tcpIpDhcpV6 CnfDelayMin	TimeValue	0..1	attr	Minimum delay in seconds before the first Confirm message will be sent.
tcpIpDhcpV6Inf DelayMax	TimeValue	0..1	attr	Maximum delay in seconds 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.
tcpIpDhcpV6Inf DelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Information Request message will be sent.
tcpIpDhcpV6Sol DelayMax	TimeValue	0..1	attr	Maximum delay in seconds 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.
tcpIpDhcpV6Sol DelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Solicit message will be sent.

Table A.304: Dhcpv6Props

Class	**DiagEventDebounceCounterBased**			
Note	This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounce CounterBased.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounce Algorithm			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
counterBased FdcThreshold StorageValue	Integer	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
counter DecrementStep Size	Integer	0..1	attr	This value shall be taken to decrement the internal debounce counter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**DiagEventDebounceCounterBased**			
counterFailed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "failed" counter status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counter IncrementStep Size	Integer	0..1	attr	This value shall be taken to increment the internal debounce counter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJump Down	Boolean	0..1	attr	This value activates or deactivates the counter jump-down behavior. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJump DownValue	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJumpUp	Boolean	0..1	attr	This value activates or deactivates the counter jump-up behavior. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJumpUp Value	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterPassed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "passed" counter status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.305: DiagEventDebounceCounterBased

Class	**DiagEventDebounceMonitorInternal**			
Note	This meta-class represents the ability to indicate that no Dem pre-debounce algorithm shall be used for this diagnostic monitor. The SWC might implement an internal debouncing algorithm and report qualified (debounced) results to the Dem/DM.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounceAlgorithm			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.306: DiagEventDebounceMonitorInternal

Class	**DiagEventDebounceTimeBased**
Note	This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTimeBase.
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	DiagEventDebounceTimeBased			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounceAlgorithm			
Attribute	Type	Mult.	Kind	Note
timeBasedFdcThresholdStorageValue	TimeValue	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
timeFailedThreshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "failed" status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
timePassedThreshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "passed" status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.307: DiagEventDebounceTimeBased

Class	***DiagnosticAbstractDataIdentifier*** (abstract)			
Note	This meta-class represents an abstract base class for the modeling of a diagnostic data identifier (DID).			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticDataIdentifier, DiagnosticDynamicDataIdentifier			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.308: DiagnosticAbstractDataIdentifier

Class	***DiagnosticAbstractParameter*** (abstract)			
Note	This meta-class represents an abstract base class for modeling a diagnostic parameter.			
Base	*ARObject*			
Subclasses	DiagnosticParameter, DiagnosticParameterElement			
Attribute	Type	Mult.	Kind	Note
bitOffset	PositiveInteger	0..1	attr	This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction. **Stereotypes:** atpIdentityContributor **Tags:** atp.Status=candidate

▽

△

Class	***DiagnosticAbstractParameter*** (abstract)			
dataElement	DiagnosticDataElement	0..1	aggr	This represents the related dataElement of the Diagnostic Parameter **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.shortName, data Element.variationPoint.shortLabel vh.latestBindingTime=postBuild
parameterSize	PositiveInteger	0..1	attr	This attribute allows for the specification of the parameter size. This information is relevant if there is a gap between one diagnostic parameter and the following diagnostic parameter (or the tail of the telegram). The unit is bit and the values shall be multiples of 8. **Tags:** atp.Status=candidate

Table A.309: DiagnosticAbstractParameter

Class	**DiagnosticAccessPermission**			
Note	This represents the specification of whether a given service can be accessed according to the existence of meta-classes referenced by a particular DiagnosticAccessPermission. In other words, this meta-class acts as a mapping element between several (otherwise unrelated) pieces of information that are put into context for the purpose of checking for access rights. **Tags:** atp.recommendedPackage=DiagnosticAccessPermissions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Enabled	DiagnosticAuthRole Proxy	0..1	aggr	The existence of this aggregation indicates that an authentication is foreseen. The details are clarified by the aggregated class. **Tags:** atp.Splitkey=authenticationEnabled
diagnostic Session	DiagnosticSession	*	ref	This represents the associated DiagnosticSessions **Tags:** atp.Splitkey=diagnosticSession
environmental Condition	Diagnostic EnvironmentalCondition	0..1	ref	This represents the environmental conditions associated with the access permission. **Tags:** atp.Splitkey=environmentalCondition
securityLevel	DiagnosticSecurityLevel	*	ref	This represents the associated DiagnosticSecurityLevels **Tags:** atp.Splitkey=securityLevel

Table A.310: DiagnosticAccessPermission

Class	**DiagnosticAging**			
Note	Defines the aging algorithm. **Tags:** atp.recommendedPackage=DiagnosticAgings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticAging**			
agingCycle	DiagnosticOperation Cycle	0..1	ref	This represents the applicable aging cycle. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=agingCycle, agingCycle.diagnosticOperation Cycle, agingCycle.variationPoint.shortLabel vh.latestBindingTime=postBuild
threshold	PositiveInteger	0..1	attr	Number of aging cycles needed to unlearn/delete the event. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.311: DiagnosticAging

Class	***DiagnosticAuthentication*** (abstract)			
Note	This meta-class represents the ability to configure the usage of the UDS service Authentication in the Diagnostic extract.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticAuthTransmitCertificate, DiagnosticAuthenticationConfiguration, DiagnosticDeAuthentication, DiagnosticProofOfOwnership, DiagnosticVerifyCertificateBidirectional, DiagnosticVerifyCertificate Unidirectional			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Class	Diagnostic AuthenticationClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Tags:** atp.Splitkey=authenticationClass

Table A.312: DiagnosticAuthentication

Class	**DiagnosticAuthenticationConfiguration**			
Note	This meta-class represents the subfunction to configure the authentication. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.313: DiagnosticAuthenticationConfiguration

Class	**DiagnosticClearResetEmissionRelatedInfo**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x04 service. **Tags:** atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearReset Emission Related DiagnosticInfo Class	DiagnosticClearReset EmissionRelatedInfo Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearReste EmissionRelatedInfo in the given context. **Tags:** atp.Splitkey=clearResetEmissionRelatedDiagnostic InfoClass

Table A.314: DiagnosticClearResetEmissionRelatedInfo

Class	**DiagnosticComControl**			
Note	This represents an instance of the "Communication Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCommunicationControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comControl Class	DiagnosticComControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticComControl in the given context. **Tags:** atp.Splitkey=comControlClass
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.

Table A.315: DiagnosticComControl

Class	**DiagnosticComControlSpecificChannel**			
Note	This represents the ability to add further attributes to the definition of a specific channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.specificChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
specificChannel	CommunicationCluster	0..1	ref	This represents the affected CommunicationCluster in the role specificChannel **Tags:** atp.Splitkey=specificChannel
specificPhysical Channel	EthernetPhysical Channel	0..1	ref	This represents the affected specific EthernetPhysical Channel. **Tags:** atp.Splitkey=specificPhysicalChannel
subnetNumber	PositiveInteger	0..1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

Table A.316: DiagnosticComControlSpecificChannel

Class	**DiagnosticComControlSubNodeChannel**			
Note	This represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.subNodeChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
subNode Channel	CommunicationCluster	0..1	ref	This represents the affected CommunicationCluster in the role subNodeChannel **Tags:** atp.Splitkey=subNodeChannel
subNode Number	PositiveInteger	0..1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).
subNode Physical Channel	EthernetPhysical Channel	0..1	ref	This represents the affected sub-node EthernetPhysical Channel. **Tags:** atp.Splitkey=subNodePhysicalChannel

Table A.317: DiagnosticComControlSubNodeChannel

Class	***DiagnosticCommonElement*** (abstract)			
Note	This meta-class represents a common base class for all diagnostic elements. It does not contribute any specific functionality other than the ability to become the target of a reference.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*DiagnosticAbstractAliasEvent*, *DiagnosticAbstractDataIdentifier*, DiagnosticAccessPermission, DiagnosticAging, DiagnosticAuthRole, *DiagnosticCondition*, *DiagnosticConditionGroup*, DiagnosticCustomServiceClass, DiagnosticDataIdentifierSet, DiagnosticEcuInstanceProps, DiagnosticEnvironmentalCondition, DiagnosticEvent, DiagnosticExtendedDataRecord, DiagnosticFimEventGroup, DiagnosticFreezeFrame, DiagnosticFunctionIdentifier, DiagnosticFunctionIdentifierInhibit, DiagnosticIndicator, DiagnosticInfoType, DiagnosticIumpr, DiagnosticIumprDenominatorGroup, DiagnosticIumprGroup, DiagnosticJ1939ExpandedFreezeFrame, DiagnosticJ1939FreezeFrame, DiagnosticJ1939Node, DiagnosticJ1939Spn, *DiagnosticMapping*, DiagnosticMeasurementIdentifier, *DiagnosticMemoryDestination*, DiagnosticMemoryIdentifier, DiagnosticOperationCycle, DiagnosticParameterIdentifier, DiagnosticPowertrainFreezeFrame, DiagnosticProtocol, DiagnosticRoutine, DiagnosticSecurityLevel, *DiagnosticServiceClass*, *DiagnosticServiceInstance*, DiagnosticServiceTable, DiagnosticSession, DiagnosticTestResult, DiagnosticTestRoutineIdentifier, *DiagnosticTroubleCode*, DiagnosticTroubleCodeGroup, DiagnosticTroubleCodeProps			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.318: DiagnosticCommonElement

Class	«atpVariation» **DiagnosticCommonProps**			
Note	This meta-class aggregates a number of common properties that are shared among a diagnostic extract. **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	*ARObject*			
Aggregated by	DiagnosticContributionSet.commonProperties			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Timeout	TimeValue	0..1	attr	This attribute defines the time (in seconds) that the authentication state is maintained in default-session if there is no communication from the authenticated client.

△

Class	«atpVariation» **DiagnosticCommonProps**			
debounce AlgorithmProps	DiagnosticDebounce AlgorithmProps	*	aggr	Defines the used debounce algorithms relevant in the context of the enclosing DiagnosticCommonProps. Usually, there is a variety of debouncing algorithms to take into account and therefore the multiplicity of this aggregation is set to 0..*. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=debounceAlgorithmProps, debounce AlgorithmProps.shortName, debounceAlgorithm Props.variationPoint.shortLabel vh.latestBindingTime=postBuild
default Endianness	ByteOrderEnum	0..1	attr	Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticData Element does not define the endianness via the swData DefProps.baseType attribute.
event Combination Reporting Behavior	DiagnosticEvent CombinationReporting BehaviorEnum	0..1	attr	In case of EventCombination on Retrieval, this attribute specifies if a specific order of reporting is to be maintained.
maxNumberOf Request Correctly Received Response Pending	PositiveInteger	0..1	attr	Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.
occurrence Counter Processing	DiagnosticOccurrence CounterProcessing Enum	0..1	attr	This attribute defines the consideration of the fault confirmation process for the occurrence counter.
resetConfirmed BitOnOverflow	Boolean	0..1	attr	This attribute defines whether the confirmed bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory.
resetPendingBit OnOverflow	Boolean	0..1	attr	This attribute defines whether the pending bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory. In order to be compliant to ISO 14229-1 [3], this parameter needs to be set to "false".
responseOnAll RequestSids	Boolean	0..1	attr	If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
responseOn Second Declined Request	Boolean	0..1	attr	Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeat Request. FALSE: when the second request (Client B) can not be processed, it shall not be responded.
typeOfEvent Combination Supported	DiagnosticEvent CombinationBehavior Enum	0..1	attr	Select type of Event Combination support.

Table A.319: DiagnosticCommonProps

Enumeration	**DiagnosticCompareTypeEnum**
Note	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
Aggregated by	*DiagnosticEnvCompareCondition*.compareType
Literal	***Description***
isEqual	equal **Tags:** atp.EnumerationLiteralIndex=0
isGreaterOrEqual	greater than or equal **Tags:** atp.EnumerationLiteralIndex=5
isGreaterThan	greater than **Tags:** atp.EnumerationLiteralIndex=4
isLessOrEqual	less than or equal **Tags:** atp.EnumerationLiteralIndex=3
isLessThan	less than **Tags:** atp.EnumerationLiteralIndex=2
isNotEqual	not equal **Tags:** atp.EnumerationLiteralIndex=1

Table A.320: DiagnosticCompareTypeEnum

Class	***DiagnosticCondition*** (abstract)			
Note	Abstract element for StorageConditions and EnableConditions.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticEnableCondition, DiagnosticStorageCondition			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initValue	Boolean	0..1	attr	Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled

Table A.321: DiagnosticCondition

Class	**DiagnosticConnectedIndicator**			
Note	Description of indicators that are defined per DiagnosticEvent.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticEvent.connectedIndicator			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
behavior	DiagnosticConnected IndicatorBehaviorEnum	0..1	attr	Behavior of the linked indicator.
healingCycle Counter Threshold	PositiveInteger	0..1	attr	This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticConnectedIndicator**			
indicator	DiagnosticIndicator	0..1	ref	Reference to the used indicator. **Tags:** atp.Splitkey=indicator
indicatorFailure CycleCounter Threshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the WarningIndicatorOnCriteria. Please note that this attribute is not relevant for the Adaptive Platform.

Table A.322: DiagnosticConnectedIndicator

Class	**DiagnosticConnection**			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. **Tags:** atp.recommendedPackage=DiagnosticConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages. **Tags:** atp.Splitkey=functionalRequest
periodic ResponseUudt	PduTriggering	*	ref	Reference to UUDT responses. **Tags:** atp.Splitkey=periodicResponseUudt
physical Request	TpConnectionIdent	0..1	ref	Reference to a physical request message. **Tags:** atp.Splitkey=physicalRequest
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. **Tags:** atp.Splitkey=response

Table A.323: DiagnosticConnection

Class	**DiagnosticContributionSet**			
Note	This meta-class represents a root node of a diagnostic extract. It bundles a given set of diagnostic model elements. The granularity of the DiagonsticContributionSet is arbitrary in order to support the aspect of decentralized configuration, i.e. different contributors can come up with an own DiagnosticContribution Set. **Tags:** atp.recommendedPackage=DiagnosticContributionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
common Properties	DiagnosticCommon Props	0..1	aggr	This attribute represents a collection of diagnostic properties that are shared among the entire Diagnostic ContributionSet. **Tags:** atp.Splitkey=commonProperties
element	DiagnosticCommon Element	*	ref	This represents a DiagnosticCommonElement considered in the context of the DiagnosticContributionSet **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.diagnosticCommon Element, element.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	DiagnosticContributionSet			
serviceTable	DiagnosticServiceTable	*	ref	This represents the collection of DiagnosticServiceTables to be considered in the scope of this Diagnostic ContributionSet. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceTable, serviceTable.diagnosticService Table, serviceTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.324: DiagnosticContributionSet

Class	DiagnosticControlEnableMaskBit			
Note	This meta-class has the ability to represent one bit in the control enable mask record.			
Base	*ARObject*			
Aggregated by	DiagnosticIOControl.controlEnableMaskBit			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bitNumber	PositiveInteger	0..1	attr	This attribute represents the bit number of the bit in the control mask record. Bit number 0 is the most significant bit (MSB) in the first byte of the CEMR in the network presentation.
controlledData Element	DiagnosticDataElement	*	ref	This reference represents the collection of Diagnostic DataElements that are controlled by this bit of the control mask record. **Tags:** atp.Splitkey=controlledDataElement

Table A.325: DiagnosticControlEnableMaskBit

Class	DiagnosticCustomServiceClass			
Note	This represents the ability to define a custom diagnostic service class and assign an ID to it. Further configuration is not foreseen from the point of view of the diagnostic extract and consequently needs to be done on the level of ECUC. **Tags:** atp.recommendedPackage=DiagnosticCustomServiceClasses			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
customService Id	PositiveInteger	0..1	attr	This attribute may only be used for the definition of custom services. The values shall not overlap with existing standardized service IDs.

Table A.326: DiagnosticCustomServiceClass

Class	**DiagnosticDataByIdentifier** (abstract)			
Note	This represents an abstract base class for all diagnostic services that access data by identifier.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticReadDataByIdentifier, DiagnosticReadScalingDataByIdentifier, DiagnosticWriteDataBy Identifier			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataIdentifier	DiagnosticAbstractData Identifier	0..1	ref	This represents the linked DiagnosticDataIdentifier. **Tags:** atp.Splitkey=dataIdentifier

Table A.327: DiagnosticDataByIdentifier

Class	**DiagnosticDataElement**			
Note	This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.			
Base	*ARObject*, *DiagnosticServiceMappingDiagTarget*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*DiagnosticAbstractParameter*.dataElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the meaning of the value of the array size.
maxNumberOf Elements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
scalingInfoSize	PositiveInteger	0..1	attr	Size in bytes of scaling information for the DiagnosticData Element if used with DiagnosticReadScalingDataBy Identifier
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties in order to support the definition of e.g. computation formulae and data constraints. **Tags:** atp.Splitkey=swDataDefProps

Table A.328: DiagnosticDataElement

Class	**DiagnosticDataIdentifier**			
Note	This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommon Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticParameter	*	aggr	This is the dataElement associated with the Diagnostic DataIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
didSize	PositiveInteger	0..1	attr	This attribute indicates the size in bytes of the Diagnostic DataIdentifier.

▽

△

Class	DiagnosticDataIdentifier			
representsVin	Boolean	0..1	attr	This attributes indicates whether the specific Diagnostic DataIdentifier represents the vehicle identification.
supportInfoByte	DiagnosticSupportInfo Byte	0..1	aggr	This attribute represents the supported information associated with the DiagnosticDataIdentifier. **Tags:** atp.Splitkey=supportInfoByte

Table A.329: DiagnosticDataIdentifier

Class	DiagnosticDataIdentifierSet			
Note	This represents the ability to define a list of DiagnosticDataIdentifiers that can be reused in different contexts. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifierSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
diagnosticData Identifier (ordered)	DiagnosticDataIdentifier	*	ref	Reference to an ordered list of Data Identifiers. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.330: DiagnosticDataIdentifierSet

Class	DiagnosticDeAuthentication			
Note	This meta-class represents the subfunction to remove the authentication **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.331: DiagnosticDeAuthentication

Class	DiagnosticDebounceAlgorithmProps			
Note	Defines properties for the debounce algorithm class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticCommonProps.debounceAlgorithmProps			
Attribute	Type	Mult.	Kind	Note
debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	This represents the actual debounce algorithm. **Tags:** atp.Splitkey=debounceAlgorithm, debounce Algorithm.shortName

▽

△

Class	DiagnosticDebounceAlgorithmProps			
debounce Behavior	DiagnosticDebounce BehaviorEnum	0..1	attr	This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
debounce CounterStorage	Boolean	0..1	attr	Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile Please note that this attribute is not relevant for the adaptive platform. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.332: DiagnosticDebounceAlgorithmProps

Class	DiagnosticDynamicDataIdentifier			
Note	This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.333: DiagnosticDynamicDataIdentifier

Class	DiagnosticDynamicallyDefineDataIdentifier			
Note	This represents an instance of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataIdentifier	DiagnosticDynamicData Identifier	0..1	ref	This represents the applicable DiagnosticDynamicData Identfier. **Tags:** atp.Splitkey=dataIdentifier
dynamically DefineData IdentifierClass	DiagnosticDynamically DefineDataIdentifier Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDynamicallyDefine DataIdentifier in the given context. **Tags:** atp.Splitkey=dynamicallyDefineDataIdentifierClass
maxSource Element	PositiveInteger	0..1	attr	This represents the maximum number of source elements of the dynamically created DID.

Table A.334: DiagnosticDynamicallyDefineDataIdentifier

Class	**DiagnosticDynamicallyDefineDataIdentifierClass**			
Note	This meta-class contains attributes shared by all instances of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
checkPer SourceId	Boolean	0..1	attr	If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0x F200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.
configuration Handling	DiagnosticHandle DDDIConfiguration Enum	0..1	attr	This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
subfunction	DiagnosticDynamically DefineDataIdentifier SubfunctionEnum	*	attr	This attribute contains a list of applicable subfunctions for all DiagnosticDynamicallyDefineDataIdentifier that reference the DiagnosticDynamicallyDefineDataIdentifier Class.

Table A.335: DiagnosticDynamicallyDefineDataIdentifierClass

Enumeration	**DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum**
Note	This meta-class contains a list of possible subfunctions for the UDS service 0x2C.
Aggregated by	DiagnosticDynamicallyDefineDataIdentifierClass.subfunction
Literal	***Description***
clearDynamically DefineDataIdentifier	Clear the specified dynamic data identifier. **Tags:** atp.EnumerationLiteralIndex=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. **Tags:** atp.EnumerationLiteralIndex=1
defineByMemory Address	The definition of dynamic data identifier shall be done via a reference to a memory address. **Tags:** atp.EnumerationLiteralIndex=2

Table A.336: DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum

Class	**DiagnosticEcuInstanceProps**			
Note	This meta-class represents the ability to model properties that are specific for a given EcuInstance but on the other hand represent purely diagnostic-related information. In the spirit of decentralized configuration it is therefore possible to specify the diagnostic-related information related to a given EcuInstance even if the EcuInstance does not yet exist. **Tags:** atp.recommendedPackage=DiagnosticEcuInstancePropss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticEcuInstanceProps**			
ecuInstance	EcuInstance	*	ref	This represents the actual EcuInstance to which the information contained in the DiagnosticEcuInstance contribute. **Tags:** atp.Splitkey=ecuInstance
obdSupport	DiagnosticObdSupport Enum	0..1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.

Table A.337: DiagnosticEcuInstanceProps

Class	**DiagnosticEcuReset**			
Note	This represents an instance of the "ECU Reset" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticEcuResets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.
ecuResetClass	DiagnosticEcuReset Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticEcuReset in the given context. **Tags:** atp.Splitkey=ecuResetClass

Table A.338: DiagnosticEcuReset

Class	**DiagnosticEdrDataProviderMapping**			
Note	This meta-class represents the definition of a data provider for the content of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticDataProviders			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataProvider	NameToken	0..1	attr	This attribute defines the kind of data that is associated with the referenced DiagnosticExtendedDataRecord Element.
extendedData RecordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the DiagnosticExtendedData RecordElement to which the data provider is associated with. **Tags:** atp.Splitkey=extendedDataRecordElement

Table A.339: DiagnosticEdrDataProviderMapping

Class	**DiagnosticEdrSenderPortMapping**
Note	This class is used to map a primitive DataPrototype owned by a PortPrototype that is typed by a Data Interface to an element of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticPortMappings

▽

△

Class	**DiagnosticEdrSenderPortMapping**			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataPrototype	DataPrototype	0..1	iref	This reference identifies the DataPrototype that shall be accessed for filling the extend data record in the context of a System. **Tags:** atp.Splitkey=dataPrototype, dataPrototype.context RootComposition, dataPrototype.contextComponent, dataPrototype.contextPort, dataPrototype.rootData Prototype, dataPrototype.contextDataPrototype, data Prototype.targetDataPrototype **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef
recordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the applicable extended data record element. **Tags:** atp.Splitkey=recordElement

Table A.340: DiagnosticEdrSenderPortMapping

Class	**DiagnosticEdrServerPortMapping**			
Note	This mapping class defines to which SWC service ports the element of the affected extended data record is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries. **Tags:** atp.Splitkey=bswServiceDependency
recordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the element of the extended data record to which the value obteained from either application software or basic software shall be assigned. **Tags:** atp.Splitkey=recordElement
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table A.341: DiagnosticEdrServerPortMapping

Class	DiagnosticEnableCondition			
Note	Specification of an enable condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.342: DiagnosticEnableCondition

Class	DiagnosticEnableConditionGroup			
Note	Enable condition group which includes one or several enable conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticConditionGroup*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enableCondition	DiagnosticEnable Condition	*	ref	Reference to enableConditions that are part of the Enable ConditionGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=enableCondition, enableCondition.diagnostic EnableCondition, enableCondition.variationPoint.short Label vh.latestBindingTime=postBuild

Table A.343: DiagnosticEnableConditionGroup

Class	DiagnosticEnableConditionPortMapping			
Note	Defines to which SWC service ports the DiagnosticEnableCondition is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enableCondition	DiagnosticEnable Condition	0..1	ref	Reference to the EnableCondition which is mapped to a SWC service port. **Tags:** atp.Splitkey=enableCondition
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. This reference can be used in early stages of the development in order to identify the SwcServiceDependency without a full System Context. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**DiagnosticEnableConditionPortMapping**			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table A.344: DiagnosticEnableConditionPortMapping

Class	**DiagnosticEnvBswModeElement**			
Note	This meta-class represents the ability to refer to a specific ModeDeclaration in the scope of a BswModule Description.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
mode	ModeDeclaration	0..1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **Tags:** atp.Splitkey=mode, mode.contextModeDeclaration Group, mode.targetMode **InstanceRef implemented by:** ModeInBswModule DescriptionInstanceRef

Table A.345: DiagnosticEnvBswModeElement

Class	***DiagnosticEnvCompareCondition*** (abstract)			
Note	DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.			
Base	*ARObject*, *DiagnosticEnvConditionFormulaPart*			
Subclasses	DiagnosticEnvDataCondition, DiagnosticEnvDataElementCondition, DiagnosticEnvModeCondition			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
compareType	DiagnosticCompare TypeEnum	0..1	attr	This attributes represents the concrete type of the comparison.

Table A.346: DiagnosticEnvCompareCondition

Class	**DiagnosticEnvConditionFormula**			
Note	A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvCondition Formula.nrcValue.			
Base	*ARObject*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part, DiagnosticEnvironmentalCondition.formula			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nrcValue	PositiveInteger	0..1	attr	This attribute represents the concrete NRC value that shall be returned if the condition fails.
op	DiagnosticLogical OperatorEnum	0..1	attr	This attribute represents the concrete operator (supported operators: and, or) of the condition formula.
part (ordered)	DiagnosticEnvCondition FormulaPart	*	aggr	This aggregation represents the collection of formula parts that can be combined by logical operators.

Table A.347: DiagnosticEnvConditionFormula

Class	**DiagnosticEnvDataCondition**			
Note	A DiagnosticEnvDataCondition is an atomic condition that compares the current value of the referenced DiagnosticDataElement with a constant value defined by the ValueSpecification. All compareTypes are supported.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compareValue	ValueSpecification	0..1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	DiagnosticDataElement	0..1	ref	This reference represents the related diagnostic data element.

Table A.348: DiagnosticEnvDataCondition

Class	**DiagnosticEnvDataElementCondition**			
Note	This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compareValue	ValueSpecification	0..1	aggr	This aggregation represents the definition of the compare value against which the value taken from the application software shall be compared.
dataPrototype	DataPrototype	0..1	iref	This instanceRef represent the ability to access a data element owned by the application software on the AUTOSAR classic platform. **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef
swDataDef Props	SwDataDefProps	0..1	aggr	Via this aggregation it is possible to describe the properties of the data that is obtained from the application for the environmental condition.

Table A.349: DiagnosticEnvDataElementCondition

Class	DiagnosticEnvModeCondition			
Note	DiagnosticEnvModeCondition are atomic condition based on the comparison of the active Mode Declaration in a ModeDeclarationGroupProtoype with the constant value of a ModeDeclaration. The formulation of this condition uses only one DiagnosticEnvElement, which contains enough information to deduce the variable part (i.e. the part that changes at runtime) as well as the constant part of the comparison. Only DiagnosticCompareTypeEnum.isEqual or DiagnosticCompareTypeEnum.isNotEqual are eligible values for DiagnosticAtomicCondition.compareType.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeElement	DiagnosticEnvMode Element	0..1	ref	This reference represents both the ModeDeclaration GroupPrototype and the ModeDeclaration relevant for the mode comparison.

Table A.350: DiagnosticEnvModeCondition

Class	***DiagnosticEnvModeElement*** (abstract)			
Note	All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition shall be defined as a DiagnosticEnvModeElement of this DiagnosticEnvironmentalCondition. This concept keeps the ARXML clean: It avoids that the DiagnosticEnvConditionFormula is cluttered by lengthy InstanceRef definitions. Furthermore, it allows that an InstanceRef only needs to be defined once and can be used multiple times in the different DiagnosticEnvModeConditions.			
Base	*ARObject*, *Referrable*			
Subclasses	DiagnosticEnvBswModeElement, DiagnosticEnvSwcModeElement			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.351: DiagnosticEnvModeElement

Class	DiagnosticEnvSwcModeElement			
Note	This meta-class represents the ability to refer to a ModeDeclaration in a concrete System context.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode	ModeDeclaration	0..1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **Tags:** atp.Splitkey=mode, mode.contextComposition, mode.contextComponent, mode.contextPPort, mode.contextModeDeclarationGroup, mode.targetMode **InstanceRef implemented by:** PModeInSystemInstance Ref

Table A.352: DiagnosticEnvSwcModeElement

Class	DiagnosticEnvironmentalCondition			
Note	The meta-class DiagnosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at "environmental" states (e.g. one such condition is that the vehicle is not driving, i.e. vehicle speed == 0). **Tags:** atp.recommendedPackage=DiagnosticEnvironmentalConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
formula	DiagnosticEnvCondition Formula	0..1	aggr	This attribute represents the formula part of the DiagnosticEnvironmentalCondition. **Tags:** atp.Splitkey=formula
modeElement	DiagnosticEnvMode Element	*	aggr	This aggregation contains a representation of Mode Declarations in the context of a DiagnosticEnvironmental Condition. **Tags:** atp.Splitkey=modeElement, modeElement.short Name

Table A.353: DiagnosticEnvironmentalCondition

Class	DiagnosticEvent			
Note	This element is used to configure DiagnosticEvents. **Tags:** atp.recommendedPackage=DiagnosticEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
associated Event Identification	PositiveInteger	0..1	attr	This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.
clearEvent Allowed Behavior	DiagnosticClearEvent AllowedBehaviorEnum	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback
confirmation Threshold	PositiveInteger	0..1	attr	This attribute defines the number of operation cycles with a failed result before a confirmed DTC is set to 1. The semantic of this attribute is a by "1" increased value compared to the confirmation threshold of the "trip counter" mentioned in ISO 14229-1 in figure D.4. A value of "1" defines the immediate confirmation of the DTC along with the first reported failed. This is also sometimes called "zero trip DTC". A value of "2" defines a DTC confirmation in the operation cycle after the first occurred failed. A value of "2" is typically used in the US for OBD DTC confirmation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
connected Indicator	DiagnosticConnected Indicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connectedIndicator, connected Indicator.shortName, connectedIndicator.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticEvent**			
eventClear Allowed	DiagnosticEventClear AllowedEnum	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback. This Attribute is only used by the AUTOSAR Classic Platform.
eventKind	DiagnosticEventKind Enum	0..1	attr	This attribute is used to distinguish between SWC and BSW events. This Attribute is only used by the AUTOSAR Classic Platform.
prestorage FreezeFrame	Boolean	0..1	attr	This attribute describes whether the Prestorage of Freeze Frames is supported by the assigned event or not. true: Prestorage of FreezeFrames is supported fFalse: Prestorage of FreezeFrames is not supported
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm)
recoverableIn SameOperation Cycle	Boolean	0..1	attr	If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.

Table A.354: DiagnosticEvent

Enumeration	**DiagnosticEventCombinationBehaviorEnum**
Note	Select type of Event Combination support
Aggregated by	DiagnosticCommonProps.typeOfEventCombinationSupported
Literal	***Description***
eventCombination OnRetrieval	Event combination on retrieval is used to combine events. For each event an individual event memory entry is created, while reporting the data via UDS, the data is combined. **Tags:** atp.EnumerationLiteralIndex=1
eventCombination OnStorage	Event combination on storage is used to combine events. Only one memory entry exists for each DTC which is also reported via UDS. **Tags:** atp.EnumerationLiteralIndex=0

Table A.355: DiagnosticEventCombinationBehaviorEnum

Enumeration	**DiagnosticEventCombinationReportingBehaviorEnum**
Note	Select reporting format of events. Applicable only for Event Combination on Retrieval.
Aggregated by	DiagnosticCommonProps.eventCombinationReportingBehavior
Literal	***Description***
reportingIn ChronologicalOrder OldestFirst	The reporting order for event combination on retrieval is the chronological storage order of the events **Tags:** atp.EnumerationLiteralIndex=1

Table A.356: DiagnosticEventCombinationReportingBehaviorEnum

Class	**DiagnosticEventNeeds**			
Note	Specifies the abstract needs on the configuration of the Diagnostic Event Manager for one diagnostic event. Its shortName can be regarded as a symbol identifying the diagnostic event from the viewpoint of the component or module which owns this element. In case the diagnostic event specifies a production error, the shortName shall be the name of the production error.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
deferringFid	FunctionInhibitionNeeds	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result. **Tags:** atp.Splitkey=deferringFid
diagEvent Debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager. **Tags:** atp.Splitkey=diagEventDebounceAlgorithm, diag EventDebounceAlgorithm.shortName
inhibitingFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for inhibition of the diagnostic monitor. The FID might either inhibit the monitoring of a symptom or the reporting of detected faults. **Tags:** atp.Splitkey=inhibitingFid
inhibiting SecondaryFid	FunctionInhibitionNeeds	*	ref	This represents the secondary Function Inhibition Identifier used for inhibition of the diagnostic monitor. Any of the FID inhibitions leads to an inhibition of the monitoring of a symptom or the reporting of detected faults. **Tags:** atp.Splitkey=inhibitingSecondaryFid
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm).
usesMonitor Data	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

Table A.357: DiagnosticEventNeeds

Class	**DiagnosticEventPortMapping**			
Note	Defines to which SWC service ports the DiagnosticEvent is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries. **Tags:** atp.Splitkey=bswServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	DiagnosticEventPortMapping			
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports. **Tags:** atp.Splitkey=diagnosticEvent
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table A.358: DiagnosticEventPortMapping

Class	DiagnosticEventToDebounceAlgorithmMapping			
Note	Defines which Debounce Algorithm is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
debounce Algorithm	DiagnosticDebounce AlgorithmProps	0..1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=debounceAlgorithm
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Debounce Algorithm is assigned. **Tags:** atp.Splitkey=diagnosticEvent

Table A.359: DiagnosticEventToDebounceAlgorithmMapping

Class	DiagnosticEventToEnableConditionGroupMapping			
Note	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which an Enable ConditionGroup is assigned. **Tags:** atp.Splitkey=diagnosticEvent

▽

△

Class	DiagnosticEventToEnableConditionGroupMapping			
enableCondition Group	DiagnosticEnable ConditionGroup	0..1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=enableConditionGroup

Table A.360: DiagnosticEventToEnableConditionGroupMapping

Class	DiagnosticEventToOperationCycleMapping			
Note	Defines which OperationCycle is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which an Operation Cycle is assigned. **Tags:** atp.Splitkey=diagnosticEvent
diagnostic OperationCycle	DiagnosticOperation Cycle	0..1	ref	Reference to an OperationCycle assigned to a Diagnostic Event. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticOperationCycle, diagnostic OperationCycle.diagnosticOperationCycle, diagnostic OperationCycle.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.361: DiagnosticEventToOperationCycleMapping

Class	DiagnosticEventToSecurityEventMapping			
Note	This meta-class represents the ability to map a security event that is defined in the context of the Security Extract to a diagnostic event defined on the context of the DiagnosticExtract. **Tags:** atp.Status=candidate atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	This reference identifies the applicable diagnostic event. **Tags:** atp.Splitkey=diagnosticEvent atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.
securityEvent Props	SecurityEventContext Props	0..1	ref	This reference identifies the qualification of the applicable security event **Tags:** atp.Splitkey=securityEventProps atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table A.362: DiagnosticEventToSecurityEventMapping

Class	DiagnosticEventToStorageConditionGroupMapping			
Note	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Storage ConditionGroup is assigned. **Tags:** atp.Splitkey=diagnosticEvent
storage ConditionGroup	DiagnosticStorage ConditionGroup	0..1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=storageConditionGroup

Table A.363: DiagnosticEventToStorageConditionGroupMapping

Class	DiagnosticEventToTroubleCodeJ1939Mapping			
Note	By means of this meta-class it is possible to associate a DiagnosticEvent to a DiagnosticTroubleCode J1939. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a J1939 Diagnostic Trouble Code is assigned. **Tags:** atp.Splitkey=diagnosticEvent
troubleCode J1939	DiagnosticTroubleCode J1939	0..1	ref	Reference to a J1939 Diagnostic Trouble Code to which a DiagnosticEvent is assigned. **Tags:** atp.Splitkey=troubleCodeJ1939

Table A.364: DiagnosticEventToTroubleCodeJ1939Mapping

Class	DiagnosticEventToTroubleCodeUdsMapping			
Note	Defines which UDS Diagnostic Trouble Code is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a UDS Diagnostic Trouble Code is assigned. **Tags:** atp.Splitkey=diagnosticEvent
troubleCodeUds	DiagnosticTroubleCode Uds	0..1	ref	Reference to an UDS Diagnostic Trouble Code assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=troubleCodeUds

Table A.365: DiagnosticEventToTroubleCodeUdsMapping

Class	DiagnosticExtendedDataRecord			
Note	Description of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticExtendedDataRecords			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
element (ordered)	DiagnosticExtended DataRecordElement	*	aggr	This aggregation represents the collection of elements of the enclosing Extentded Data Record.
recordNumber	PositiveInteger	0..1	attr	This attribute specifies an unique identifier for an extended data record.
trigger	DiagnosticRecord TriggerEnum	0..1	attr	This attribute specifies the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines when an extended data record is captured. true: This extended data record is captured every time. false: This extended data record is only captured for new event memory entries.

Table A.366: DiagnosticExtendedDataRecord

Class	DiagnosticExtendedDataRecordElement			
Note	This meta-class represents an element of an extended data record. Such an element represents a primitive data object or a fixed-size array of primitive data objects without further internal structure (no structure allowed).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticExtendedDataRecord.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
numberOf Elements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into a fixed-size array of data. The attribute determines the size of the array in terms of how many elements the array shall take. The aggregation of SwDataDefProps (especially with respect to SwBaseType) applies specifically to an element of the array (as a representation of all array elements) instead of the entire array itself!
swDataDef Props	SwDataDefProps	0..1	aggr	This aggregation allows for the specification of various properties for the enclosing element. **Tags:** atp.Splitkey=swDataDefProps

Table A.367: DiagnosticExtendedDataRecordElement

Class	DiagnosticFimAliasEventGroupMapping
Note	This meta-class represents the ability to map a DiagnosticFimEventGroup to a DiagnosticFimAliasEvent Group. By this means the "preliminary" modeling by way of a DiagnosticFimAliasEventGroup is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEventGroupMappings
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	DiagnosticFimAliasEventGroupMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
actualEvent	DiagnosticFimEvent Group	0..1	ref	This represents the reference to the actual summary event. **Tags:** atp.Splitkey=actualEvent
aliasEvent	DiagnosticFimAlias EventGroup	0..1	ref	This represents the reference to the alias summary event. **Tags:** atp.Splitkey=aliasEvent

Table A.368: DiagnosticFimAliasEventGroupMapping

Class	DiagnosticFimAliasEventMapping			
Note	This meta-class represents the ability to model the mapping of a DiagnosticEvent to a DiagnosticAlias Event. By this means the "preliminary" modeling by way of a DiagnosticAliasEvent is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimEventMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
actualEvent	DiagnosticEvent	0..1	ref	This represents the reference to the actual diagnostic event. **Tags:** atp.Splitkey=actualEvent
aliasEvent	DiagnosticFimAlias Event	0..1	ref	This represents the reference to the alias event. **Tags:** atp.Splitkey=aliasEvent

Table A.369: DiagnosticFimAliasEventMapping

Class	DiagnosticFimFunctionMapping			
Note	This meta-class represents the ability to define a mapping between a function identifier (FID) and the corresponding SwcServiceDependency in the application software resp. basic software. **Tags:** atp.recommendedPackage=DiagnosticFimFunctionMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **Tags:** atp.Splitkey=mappedBswServiceDependency
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy. **Tags:** atp.Splitkey=mappedFlatSwcServiceDependency
mapped Function	DiagnosticFunction Identifier	0..1	ref	This represents the mapped FID. **Tags:** atp.Splitkey=mappedFunction

▽

△

Class	**DiagnosticFimFunctionMapping**			
mappedSwc Service Dependency	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition). **Tags:** atp.Splitkey=mappedSwcServiceDependency, mappedSwcServiceDependency.base, mappedSwc ServiceDependency.contextRootSwComposition, mappedSwcServiceDependency.contextSwComponent Prototype, mappedSwcServiceDependency.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table A.370: DiagnosticFimFunctionMapping

Class	**DiagnosticFreezeFrame**			
Note	This element describes combinations of DIDs for a non OBD relevant freeze frame. **Tags:** atp.recommendedPackage=DiagnosticFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
recordNumber	PositiveInteger	0..1	attr	This attribute defines a record number for a freeze frame record. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
trigger	DiagnosticRecord TriggerEnum	0..1	attr	This attribute defines the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines the approach when the freeze frame record is stored/updated. true: FreezeFrame record is captured every time. false: FreezeFrame record is only captured for new event memory entries.

Table A.371: DiagnosticFreezeFrame

Class	**DiagnosticFunctionIdentifierInhibit**			
Note	This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration. **Tags:** atp.recommendedPackage=DiagnosticFunctionIdentifierInhibits			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
function Identifier	DiagnosticFunction Identifier	0..1	ref	This represents the corresponding function identifier. **Tags:** atp.Splitkey=functionIdentifier
inhibitionMask	DiagnosticInhibition MaskEnum	0..1	attr	This represents the value of the inhibition mask behavior.

▽

△

Class	DiagnosticFunctionIdentifierInhibit			
inhibitSource	DiagnosticFunction InhibitSource	*	aggr	This represents a collection of DiagnosticFunctionInhibit Source that contribute to the configuration of the enclosing DiagnosticFunctionIdentifierInhibit. **Tags:** atp.Splitkey=inhibitSource, inhibitSource.short Name

Table A.372: DiagnosticFunctionIdentifierInhibit

Class	DiagnosticFunctionInhibitSource			
Note	This meta-class represents the ability to define an inhibition source in the context of the Fim configuration.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticFunctionIdentifierInhibit.inhibitSource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
event	DiagnosticFimAlias Event	0..1	ref	This represents the alias event applicable for the referencing inhibition source. **Tags:** atp.Splitkey=event
eventGroup	DiagnosticFimAlias EventGroup	0..1	ref	This represents the event group applicable for the referencing inhibition source. **Tags:** atp.Splitkey=eventGroup

Table A.373: DiagnosticFunctionInhibitSource

Class	DiagnosticIOControl			
Note	This represents an instance of the "I/O Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticIoControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
controlEnable MaskBit	DiagnosticControl EnableMaskBit	*	aggr	This aggregation represents the control mask record consisting of single bits. **Tags:** atp.Splitkey=controlEnableMaskBit
dataIdentifier	DiagnosticDataIdentifier	0..1	ref	This represents the corresponding DiagnosticData Identifier **Tags:** atp.Splitkey=dataIdentifier
freezeCurrent State	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	DiagnosticIoControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticIOControl in the given context. **Tags:** atp.Splitkey=ioControlClass
resetToDefault	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTerm Adjustment	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

Table A.374: DiagnosticIOControl

Class	DiagnosticIndicator			
Note	Definition of an indicator. **Tags:** atp.recommendedPackage=DiagnosticIndicators			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	DiagnosticIndicatorTypeEnum	0..1	attr	Defines the type of the indicator. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.375: DiagnosticIndicator

Class	DiagnosticInfoType			
Note	This meta-class represents the ability to model an OBD info type. **Tags:** atp.recommendedPackage=DiagnosticInfoTypes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticParameter	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, dataElement.ident.shortName
id	PositiveInteger	0..1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

Table A.376: DiagnosticInfoType

Class	DiagnosticIoControlNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics. **Tags:** atp.Splitkey=currentValue
freezeCurrentStateSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporary freezing of I/O value.
resetToDefaultSupported	Boolean	0..1	attr	This represents a flag for the existence of the ResetTo Default operation in the service interface.
shortTermAdjustmentSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.

Table A.377: DiagnosticIoControlNeeds

Class	DiagnosticIumprGroup			
Note	This meta-class represents the ability to model a IUMPR groups. **Tags:** atp.recommendedPackage=DiagnosticIumprGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a Diagnostic IumprGroup. **Tags:** atp.Splitkey=iumpr
iumprGroup Identifier	DiagnosticIumprGroup Identifier	0..1	aggr	This aggregation allows for the variant modeling of the groupIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iumprGroupIdentifier, iumprGroup Identifier.groupId, iumprGroupIdentifier.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.378: DiagnosticIumprGroup

Class	DiagnosticIumprGroupIdentifier			
Note	This meta-class provides the ability to the define the group identifier for an IumprGroup.			
Base	*ARObject*			
Aggregated by	DiagnosticIumprGroup.iumprGroupIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
groupId	NameToken	0..1	attr	This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName. **Stereotypes:** atpIdentityContributor

Table A.379: DiagnosticIumprGroupIdentifier

Class	DiagnosticIumprToFunctionIdentifierMapping			
Note	This meta-class represents the ability to associate a DiagnosticFunctionIdentifier with a DiagnosticIumpr. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
function Identifier	DiagnosticFunction Identifier	0..1	ref	This reference identifies the applicable Diagnostic FunctionIdentifier. **Tags:** atp.Splitkey=functionIdentifier
iumpr	DiagnosticIumpr	0..1	ref	This reference identifies the applicable DiagnosticIumpr. **Tags:** atp.Splitkey=iumpr

Table A.380: DiagnosticIumprToFunctionIdentifierMapping

Class	DiagnosticJ1939Node			
Note	This meta-class represents the diagnostic configuration of a J1939 Nm node, which in turn represents a "virtual Ecu" on the J1939 communication bus. **Tags:** atp.recommendedPackage=DiagnosticJ1939Nodes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmNode	J1939NmNode	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated. **Tags:** atp.Splitkey=nmNode

Table A.381: DiagnosticJ1939Node

Class	DiagnosticJ1939Spn			
Note	This meta-class represents the ability to model a J1939 Suspect Parameter Number (SPN). **Tags:** atp.recommendedPackage=DiagnosticJ1939Spns This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
spn	PositiveInteger	0..1	attr	This attribute represents the concrete numerical identification for the enclosing SPN.

Table A.382: DiagnosticJ1939Spn

Class	DiagnosticMasterToSlaveEventMapping			
Note	This meta-class provides the ability to map a master diagnostic event with a slave diagnostic event such that reporting of the master event with a given value also reports the slave event with the same value **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
masterEvent	DiagnosticEvent	0..1	ref	This represents the master diagnostic event. **Tags:** atp.Splitkey=masterEvent
slaveEvent	DiagnosticEvent	0..1	ref	This represents the slave diagnostic event. **Tags:** atp.Splitkey=slaveEvent

Table A.383: DiagnosticMasterToSlaveEventMapping

Class	DiagnosticMeasurementIdentifier
Note	This meta-class represents the ability to describe a measurement identifier. **Tags:** atp.recommendedPackage=DiagnosticMeasurementIdentifiers
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	DiagnosticMeasurementIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
obdMid	PositiveInteger	0..1	attr	This represents the numerical measurement Id **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.384: DiagnosticMeasurementIdentifier

Class	***DiagnosticMemoryAddressableRangeAccess*** (abstract)			
Note	This abstract base class			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticReadMemoryByAddress, DiagnosticRequestDownload, DiagnosticRequestUpload, DiagnosticWriteMemoryByAddress			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memoryRange	DiagnosticMemoryIdentifier	*	ref	This represents the formal description of the memory segment to which the DiagnosticMemoryByAddress applies. **Tags:** atp.Splitkey=memoryRange

Table A.385: DiagnosticMemoryAddressableRangeAccess

Class	***DiagnosticMemoryDestination*** (abstract)			
Note	This abstract meta-class represents a possible memory destination for a diagnostic event.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticMemoryDestinationPrimary, DiagnosticMemoryDestinationUserDefined			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
agingRequiresTestedCycle	Boolean	0..1	attr	Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
clearDtcLimitation	DiagnosticClearDtcLimitationEnum	0..1	attr	Defines the scope of the DEM_ClearDTC Api. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
dtcStatusAvailabilityMask	PositiveInteger	0..1	attr	Mask for the supported DTC status bits by the Dem.
eventDisplacementStrategy	DiagnosticEventDisplacementStrategyEnum	0..1	attr	This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.
maxNumberOfEventEntries	PositiveInteger	0..1	attr	This attribute fixes the maximum number of event entries in the fault memory.
memoryEntryStorageTrigger	DiagnosticMemoryEntryStorageTriggerEnum	0..1	attr	Describes the trigger to allocate an event memory entry.

▽

△

Class	***DiagnosticMemoryDestination*** (abstract)			
statusBit HandlingTest FailedSinceLast Clear	DiagnosticStatusBit HandlingTestFailed SinceLastClearEnum	0..1	attr	This attribute defines, whether the aging and displacement mechanism shall be applied to the "Test FailedSinceLastClear" status bits. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
typeOfFreeze FrameRecord Numeration	DiagnosticTypeOf FreezeFrameRecord NumerationEnum	0..1	attr	This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.

Table A.386: DiagnosticMemoryDestination

Class	**DiagnosticMemoryDestinationPrimary**			
Note	This represents a primary memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory Destination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
typeOfDtc Supported	DiagnosticTypeOfDtc SupportedEnum	0..1	attr	This attribute defines the format returned by Dem_Dcm GetTranslationType and does not relate to/influence the supported Dem functionality.

Table A.387: DiagnosticMemoryDestinationPrimary

Class	**DiagnosticMemoryDestinationUserDefined**			
Note	This represents a user-defined memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory Destination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
affectedBy Control DTCSetting	Boolean	0..1	attr	This attribute configures how the content of the memory is affected by an active ControlDTCSetting or not: • If the attribute is set to **true**, the user-defined fault memory is **not** updated if ControlDTCSetting is off. • If the attribute is set to **false**, the user defined fault memory is updated even if ControlDTCSetting is off.
authentication Enabled	DiagnosticAuthRole Proxy	0..1	aggr	The existence of this aggregation indicates that an authentication is foreseen. The details are clarified by the aggregated class. **Tags:** atp.Splitkey=authenticationEnabled
memoryId	PositiveInteger	0..1	attr	This represents the identifier of the user-defined memory.

Table A.388: DiagnosticMemoryDestinationUserDefined

Class	**DiagnosticMemoryIdentifier**			
Note	This meta-class represents the ability to define memory properties from the diagnostics point of view. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
access Permission	DiagnosticAccessPermission	0..1	ref	This represents that access permission defined for the specific DiagnosticMemoryIdentifier. **Tags:** atp.Splitkey=accessPermission
id	PositiveInteger	0..1	attr	This represents the identification of the memory segment.
memoryHighAddress	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHighAddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLowAddress	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLowAddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

Table A.389: DiagnosticMemoryIdentifier

Class	**DiagnosticOperationCycle**			
Note	Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling. **Tags:** atp.recommendedPackage=DiagnosticOperationCycles			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	DiagnosticOperationCycleTypeEnum	0..1	attr	Operation cycles types for the Dem.

Table A.390: DiagnosticOperationCycle

Class	**DiagnosticOperationCyclePortMapping**			
Note	Defines to which SWC service ports the DiagnosticOperationCycle is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operationCycle	DiagnosticOperationCycle	0..1	ref	Reference to the DiagnosticOperationCycle that is assigned to SWC service ports. **Tags:** atp.Splitkey=operationCycle
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	**DiagnosticOperationCyclePortMapping**			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table A.391: DiagnosticOperationCyclePortMapping

Class	**DiagnosticParameter**			
Note	This meta-class represents the ability to describe information relevant for the execution of a specific diagnostic service, i.e. it can be taken to parameterize the service.			
Base	*ARObject*, *DiagnosticAbstractParameter*			
Aggregated by	DiagnosticDataIdentifier.dataElement, DiagnosticInfoType.dataElement, DiagnosticParameter Identifier.dataElement, DiagnosticRequestRoutineResults.request, DiagnosticRequestRoutine Results.response, DiagnosticStartRoutine.request, DiagnosticStartRoutine.response, DiagnosticStop Routine.request, DiagnosticStopRoutine.response			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ident	DiagnosticParameter Ident	0..1	aggr	The aggregation in the role ident provides the ability to make the DiagnosticAbstractParameter identifiable. From the semantical point of view, the AbstractDiagnostic Parameter is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let AbstractDiagnostic Parameter directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor
supportInfo	DiagnosticParameter SupportInfo	0..1	aggr	This attribute represents the ability to define which bit of the support info byte is representing this part of the PID. **Tags:** atp.Splitkey=supportInfo

Table A.392: DiagnosticParameter

Class	**DiagnosticParameterElement**			
Note	This meta-class represents an element of a DiagnosticParameter if the DiagnosticParameter represents a structure.			
Base	*ARObject*, *DiagnosticAbstractParameter*, *DiagnosticServiceMappingDiagTarget*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticParameterElement.subElement, DiagnosticParameterIdent.subElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arraySize	PositiveInteger	0..1	attr	This attribute indicates that the enclosing Diagnostic ParameterElement represents an array and configures the array size in terms of the number of elements of the array.

▽

△

Class	DiagnosticParameterElement			
subElement	DiagnosticParameter Element	*	aggr	This collection represents the sub-elements on the next lower level. **Tags:** atp.Splitkey=subElement, subElement.shortName, subElement.bitOffset

Table A.393: DiagnosticParameterElement

Class	DiagnosticParameterElementAccess			
Note	This meta-class acts as a single point for defining structured references to a specific Diagnostic ParameterElement.			
Base	*ARObject*			
Aggregated by	DiagnosticServiceDataMapping.parameterElementAccess, DiagnosticServiceSwMapping.parameter ElementAccess			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextElement (ordered)	DiagnosticParameter Element	*	ref	This represents the context of an applicable payload that corresponds to the referenced DataPrototype in the role mappedDataElement. **Tags:** xml.sequenceOffset=10
targetElement	DiagnosticParameter Element	0..1	ref	This represents the target reference of an applicable payload that corresponds to the referenced Data Prototype in the role mappedDataElement. **Tags:** atp.Splitkey=targetElement xml.sequenceOffset=20

Table A.394: DiagnosticParameterElementAccess

Class	DiagnosticParameterIdent			
Note	This meta-class has been created to introduce the ability to become referenced into the meta-class AbstractDiagnosticParameter without breaking backwards compatibility.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *DiagnosticServiceMappingDiagTarget*, *Ident Caption*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, DiagnosticParameter.ident			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
subElement	DiagnosticParameter Element	*	aggr	This collection represents the subElements on the top level. **Tags:** atp.Splitkey=subElement, subElement.shortName, subElement.bitOffset

Table A.395: DiagnosticParameterIdent

Class	DiagnosticParameterIdentifier			
Note	This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD). **Tags:** atp.recommendedPackage=DiagnosticParameterIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	DiagnosticParameterIdentifier			
dataElement	DiagnosticParameter	*	aggr	This represents the data carried by the Diagnostic ParameterIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
pidSize	PositiveInteger	0..1	attr	The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.
supportInfoByte	DiagnosticSupportInfo Byte	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier. **Tags:** atp.Splitkey=supportInfoByte

Table A.396: DiagnosticParameterIdentifier

Class	DiagnosticPeriodicRate			
Note	This represents the ability to define a periodic rate for the specification of the "read data by periodic ID" diagnostic service.			
Base	*ARObject*			
Aggregated by	DiagnosticReadDataByPeriodicIDClass.periodicRate			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
period	TimeValue	0..1	attr	This represents the period of the DiagnosticPeriodicRate in seconds.
periodicRate Category	DiagnosticPeriodicRate CategoryEnum	0..1	attr	This attribute represents the category of the periodic rate.

Table A.397: DiagnosticPeriodicRate

Class	DiagnosticProofOfOwnership			
Note	This meta-class represents the subfunction to provide proof of ownership. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.398: DiagnosticProofOfOwnership

Class	DiagnosticProtocol
Note	This meta-class represents the ability to define a diagnostic protocol. **Tags:** atp.recommendedPackage=DiagnosticProtocols
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*

▽

△

Class	**DiagnosticProtocol**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic Connection	DiagnosticConnection	*	ref	This represents the collection of applicable Diagnostic Connections for this DiagnosticProtocol. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticConnection, diagnostic Connection.diagnosticConnection, diagnostic Connection.variationPoint.shortLabel vh.latestBindingTime=postBuild
priority	PositiveInteger	0..1	attr	This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: • 0 - Highest protocol priority • 255 - Lowest protocol priority **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.
sendRespPend OnTransToBoot	Boolean	0..1	attr	The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false"). **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
serviceTable	DiagnosticServiceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceTable, serviceTable.diagnosticService Table, serviceTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.399: DiagnosticProtocol

Class	**DiagnosticReadDataByIdentifier**			
Note	This represents an instance of the "Read Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
readClass	DiagnosticReadDataBy IdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataBy Identifier in the given context. **Tags:** atp.Splitkey=readClass

Table A.400: DiagnosticReadDataByIdentifier

Class	**DiagnosticReadDataByPeriodicID**			
Note	This represents an instance of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
readDataClass	DiagnosticReadDataBy PeriodicIDClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataBy PeriodicID in the given context. **Tags:** atp.Splitkey=readDataClass

Table A.401: DiagnosticReadDataByPeriodicID

Class	**DiagnosticReadDataByPeriodicIDClass**			
Note	This meta-class contains attributes shared by all instances of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxPeriodicDid ToRead	PositiveInteger	0..1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	DiagnosticPeriodicRate	*	aggr	This represents the description of a collection of periodic rates in which the service can be executed. **Tags:** atp.Splitkey=periodicRate
schedulerMax Number	PositiveInteger	0..1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

Table A.402: DiagnosticReadDataByPeriodicIDClass

Class	**DiagnosticReadScalingDataByIdentifier**			
Note	This represents an instance of the "Read Scaling Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
readScaling DataClass	DiagnosticReadScaling DataByIdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadScalingData ByIdentifier in the given context. **Tags:** atp.Splitkey=readScalingDataClass

Table A.403: DiagnosticReadScalingDataByIdentifier

Enumeration	**DiagnosticRecordTriggerEnum**
Note	Triggers to allocate an event memory entry.
Aggregated by	DiagnosticExtendedDataRecord.trigger, DiagnosticFreezeFrame.trigger
Literal	***Description***
confirmed	capture on "Confirmed" **Tags:** atp.EnumerationLiteralIndex=0
custom	implement custom capture **Tags:** atp.EnumerationLiteralIndex=4
fdcThreshold	capture on "FDC Threshold" **Tags:** atp.EnumerationLiteralIndex=1
pending	capture on "Pending" **Tags:** atp.EnumerationLiteralIndex=2
testFailed	capture on "Test Failed" **Tags:** atp.EnumerationLiteralIndex=3
testFailedThis OperationCycle	Test Failed This Operation Cycle. **Tags:** atp.EnumerationLiteralIndex=5 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
testPassed	Capture on testFailed bit transition 1 --> 0. **Tags:** atp.EnumerationLiteralIndex=6 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.

Table A.404: DiagnosticRecordTriggerEnum

Class	**DiagnosticRequestControlOfOnBoardDevice**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x08 service. **Tags:** atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requestControl OfOnBoard DeviceClass	DiagnosticRequest ControlOfOnBoard DeviceClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestControlOf OnBoardDevice in the given context. **Tags:** atp.Splitkey=requestControlOfOnBoardDevice Class
testId	DiagnosticTestRoutine Identifier	0..1	ref	This represents the test Id for the mode 0x08. **Tags:** atp.Splitkey=testId

Table A.405: DiagnosticRequestControlOfOnBoardDevice

Class	**DiagnosticRequestCurrentPowertrainData**
Note	This meta-class represents the ability to model an instance of the OBD mode 0x01 service. **Tags:** atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**DiagnosticRequestCurrentPowertrainData**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pid	DiagnosticParameter Identifier	0..1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service. **Tags:** atp.Splitkey=pid
requestCurrent Powertrain DiagnosticData Class	DiagnosticRequest CurrentPowertrainData Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestCurrent PowertrainData in the given context. **Tags:** atp.Splitkey=requestCurrentPowertrainDiagnostic DataClass

Table A.406: DiagnosticRequestCurrentPowertrainData

Class	**DiagnosticRequestDownload**			
Note	This represents an instance of the "Request Download" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory AddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request DownloadClass	DiagnosticRequest DownloadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestDownload in the given context. **Tags:** atp.Splitkey=requestDownloadClass

Table A.407: DiagnosticRequestDownload

Class	**DiagnosticRequestEmissionRelatedDTC**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x03/0x07 service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request Emission RelatedDtc Class	DiagnosticRequest EmissionRelated DTCClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmission RelatedDTC in the given context. **Tags:** atp.Splitkey=requestEmissionRelatedDtcClass

Table A.408: DiagnosticRequestEmissionRelatedDTC

Class	DiagnosticRequestEmissionRelatedDTCPermanentStatus			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x0A service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request EmissionRelatedDtcClass PermanentStatus	DiagnosticRequestEmissionRelatedDTCPermanentStatusClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmissionRelatedDTCPermanentStatus in the given context. **Tags:** atp.Splitkey=requestEmissionRelatedDtcClassPermanentStatus

Table A.409: DiagnosticRequestEmissionRelatedDTCPermanentStatus

Class	DiagnosticRequestOnBoardMonitoringTestResults			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x06 service. **Tags:** atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticTestResult	DiagnosticTestResult	*	ref	This reference identifies the applicable collection of test identifiers for setting up a request message for mode 0x06. **Tags:** atp.Splitkey=diagnosticTestResult
requestOnBoardMonitoringTestResultsClass	DiagnosticRequestOnBoardMonitoringTestResultsClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestOnBoardMonitoringTestResults in the given context. **Tags:** atp.Splitkey=requestOnBoardMonitoringTestResultsClass

Table A.410: DiagnosticRequestOnBoardMonitoringTestResults

Class	DiagnosticRequestPowertrainFreezeFrameData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x02 service. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
freezeFrame	DiagnosticPowertrainFreezeFrame	0..1	ref	This represents the associated freeze-frame. **Tags:** atp.Splitkey=freezeFrame

▽

△

Class	**DiagnosticRequestPowertrainFreezeFrameData**			
request Powertrain FreezeFrame Data	DiagnosticRequest PowertrainFreezeFrame DataClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestPowertrain FreezeFrameData in the given context. **Tags:** atp.Splitkey=requestPowertrainFreezeFrameData

Table A.411: DiagnosticRequestPowertrainFreezeFrameData

Class	**DiagnosticRequestRoutineResults**			
Note	This meta-class represents the ability to define the result of a diagnostic routine execution.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.requestResult			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table A.412: DiagnosticRequestRoutineResults

Class	**DiagnosticRequestUpload**			
Note	This represents an instance of the "Request Upload" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory AddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requestUpload Class	DiagnosticRequest UploadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestUpload in the given context. **Tags:** atp.Splitkey=requestUploadClass

Table A.413: DiagnosticRequestUpload

Class	**DiagnosticRequestVehicleInfo**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x09 service. **Tags:** atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticRequestVehicleInfo**			
infoType	DiagnosticInfoType	0..1	ref	This represents the info type associated with the mode 0x09 service. **Tags:** atp.Splitkey=infoType
requestVehicle Information Class	DiagnosticRequest VehicleInfoClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequesVehicleInfo in the given context. **Tags:** atp.Splitkey=requestVehicleInformationClass

Table A.414: DiagnosticRequestVehicleInfo

Class	**DiagnosticResponseOnEventClass**			
Note	This represents the ability to define common properties for all instances of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
maxNumberOf Stored DTCStatus ChangedEvents	PositiveInteger	0..1	attr	The maximum number of DTCs that can be stored as DTCs with change status within one ResponseOnEvent SchedulerRate interval.
maxNum ChangeOfData IdentfierEvents	PositiveInteger	0..1	attr	The maximum number of events that can be simultaneously configured with sub function onChangeOf DataIdentifier.
maxNum ComparisionOf ValueEvents	PositiveInteger	0..1	attr	The maximum number of events that can be simultaneously configured with sub function on ComparisonOfValues.
maxSupported DIDLength	PositiveInteger	0..1	attr	The maximum number of measurable data bytes allowed for each DID that is used for comparison or data change.
responseOn EventScheduler Rate	TimeValue	0..1	attr	The call rate of the periodic scheduler to compare the values of the DataIdentifier (DID) or to detect DTC status changes.
storeEvent Enabled	Boolean	0..1	attr	Specifies if the storeEvent functionality of the Response OnEvent diagnostic service shall be supported or not. If set to true, the storeEvent functionality is available. If set to false the storeEvent functionality is not available.

Table A.415: DiagnosticResponseOnEventClass

Class	**DiagnosticRoutine**			
Note	This meta-class represents the ability to define a diagnostic routine. **Tags:** atp.recommendedPackage=DiagnosticRoutines			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**DiagnosticRoutine**			
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
requestResult	DiagnosticRequest RoutineResults	0..1	aggr	This represents the ability to request the result of a running routine. **Tags:** atp.Splitkey=requestResult, requestResult.short Name
routineInfo	PositiveInteger	0..1	attr	This represents the routine info byte. The info byte contains a manufacturer-specific value (for the identification of record identifiers) that is reported to the tester. Other use cases for this attribute are mentioned in ISO 27145 and ISO 26021.
start	DiagnosticStartRoutine	0..1	aggr	This represents the ability to start a routine **Tags:** atp.Splitkey=start, start.shortName
stop	DiagnosticStopRoutine	0..1	aggr	This represents the ability to stop a running routine. **Tags:** atp.Splitkey=stop, stop.shortName

Table A.416: DiagnosticRoutine

Class	**DiagnosticRoutineControl**			
Note	This represents an instance of the "Routine Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRoutineControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
routine	DiagnosticRoutine	0..1	ref	This refers to the applicable DiagnosticRoutine. **Tags:** atp.Splitkey=routine
routineControl Class	DiagnosticRoutine ControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRoutineControl in the given context. **Tags:** atp.Splitkey=routineControlClass

Table A.417: DiagnosticRoutineControl

Class	**DiagnosticRoutineNeeds**			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagRoutine Type	DiagnosticRoutineType Enum	0..1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.

Table A.418: DiagnosticRoutineNeeds

Enumeration	**DiagnosticRoutineTypeEnum**
Note	This enumerator specifies the different types of diagnostic routines.
Aggregated by	DiagnosticRoutineNeeds.diagRoutineType
Literal	***Description***
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. **Tags:** atp.EnumerationLiteralIndex=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. **Tags:** atp.EnumerationLiteralIndex=1

Table A.419: DiagnosticRoutineTypeEnum

Class	**DiagnosticSecurityAccess**			
Note	This represents an instance of the "Security Access" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requestSeedId	PositiveInteger	0..1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId
securityAccessClass	DiagnosticSecurityAccessClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSecurityAccess in the given context. **Tags:** atp.Splitkey=securityAccessClass
securityDelayTimeOnBoot	TimeValue	0..1	attr	Start delay timer on power on in seconds. This delay indicates the time after ECU boot power-on where no security access request is accepted.
securityLevel	DiagnosticSecurityLevel	0..1	ref	This reference identifies the applicable security level for the security access. **Tags:** atp.Splitkey=securityLevel

Table A.420: DiagnosticSecurityAccess

Class	**DiagnosticSecurityLevel**			
Note	This meta-class represents the ability to define a security level considered for diagnostic purposes. **Tags:** atp.recommendedPackage=DiagnosticSecurityLevels			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessDataRecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	0..1	attr	This represents the size of the security key. Unit: byte.
numFailedSecurityAccess	PositiveInteger	0..1	attr	This represents the number of failed security accesses after which the delay time is activated.

△

Class	DiagnosticSecurityLevel			
securityDelay Time	TimeValue	0..1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	0..1	attr	This represents the size of the security seed. Unit: byte.

Table A.421: DiagnosticSecurityLevel

Class	*DiagnosticServiceClass* (abstract)			
Note	This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticAuthenticationClass, DiagnosticClearDiagnosticInformationClass, DiagnosticClearResetEmissionRelatedInfoClass, DiagnosticComControlClass, DiagnosticControlDTCSettingClass, DiagnosticCustomServiceClass, DiagnosticDataTransferClass, DiagnosticDynamicallyDefineDataIdentifierClass, DiagnosticEcuResetClass, DiagnosticIoControlClass, DiagnosticReadDTCInformationClass, DiagnosticReadDataByIdentifierClass, DiagnosticReadDataByPeriodicIDClass, DiagnosticReadMemoryByAddressClass, DiagnosticReadScalingDataByIdentifierClass, DiagnosticRequestControlOfOnBoardDeviceClass, DiagnosticRequestCurrentPowertrainDataClass, DiagnosticRequestDownloadClass, DiagnosticRequestEmissionRelatedDTCClass, DiagnosticRequestEmissionRelatedDTCPermanentStatusClass, DiagnosticRequestFileTransferClass, DiagnosticRequestOnBoardMonitoringTestResultsClass, DiagnosticRequestPowertrainFreezeFrameDataClass, DiagnosticRequestUploadClass, DiagnosticRequestVehicleInfoClass, DiagnosticResponseOnEventClass, DiagnosticRoutineControlClass, DiagnosticSecurityAccessClass, DiagnosticSessionControlClass, DiagnosticTransferExitClass, DiagnosticWriteDataByIdentifierClass, DiagnosticWriteMemoryByAddressClass			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.422: DiagnosticServiceClass

Class	DiagnosticServiceDataMapping			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in ServiceInterfaces on the adaptive platform. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element or (in case of a usage on the adaptive platform) mappedApDataElement. **Tags:** atp.Splitkey=diagnosticDataElement
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element. **Tags:** atp.Splitkey=diagnosticParameter xml.sequenceOffset=20

▽

△

Class	**DiagnosticServiceDataMapping**			
mappedData Element	DataPrototype	0..1	iref	This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform. **Tags:** atp.Splitkey=mappedDataElement, mappedData Element.contextRootComposition, mappedData Element.contextComponent, mappedData Element.contextPort, mappedDataElement.rootData Prototype, mappedDataElement.contextDataPrototype, mappedDataElement.targetDataPrototype **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef This Attribute is only used by the AUTOSAR Classic Platform.
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element. **Tags:** atp.Splitkey=parameterElementAccess

Table A.423: DiagnosticServiceDataMapping

Class	***DiagnosticServiceInstance*** (abstract)			
Note	This represents a concrete instance of a diagnostic service.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*DiagnosticAuthentication*, DiagnosticClearDiagnosticInformation, DiagnosticClearResetEmissionRelatedInfo, DiagnosticComControl, DiagnosticControlDTCSetting, DiagnosticCustomServiceInstance, *DiagnosticDataByIdentifier*, DiagnosticDynamicallyDefineDataIdentifier, DiagnosticEcuReset, DiagnosticIOControl, *DiagnosticMemoryByAddress*, DiagnosticReadDTCInformation, DiagnosticReadDataByPeriodicID, DiagnosticRequestControlOfOnBoardDevice, DiagnosticRequestCurrentPowertrainData, DiagnosticRequestEmissionRelatedDTC, DiagnosticRequestEmissionRelatedDTCPermanentStatus, DiagnosticRequestFileTransfer, DiagnosticRequestOnBoardMonitoringTestResults, DiagnosticRequestPowertrainFreezeFrameData, DiagnosticRequestVehicleInfo, DiagnosticResponseOnEvent, DiagnosticRoutineControl, DiagnosticSecurityAccess, DiagnosticSessionControl			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
access Permission	DiagnosticAccess Permission	0..1	ref	This represents the collection of DiagnosticAccess Permissions that allow for the execution of the referencing DiagnosticServiceInstance.. **Tags:** atp.Splitkey=accessPermission
serviceClass	DiagnosticServiceClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Stereotypes:** atpAbstract

Table A.424: DiagnosticServiceInstance

Class	***DiagnosticServiceMappingDiagTarget*** (abstract)			
Note	This meta-class serves as a base class for diagnostics-related targets of subclasses of DiagnosticSw Mapping			
Base	*ARObject*			
Subclasses	DiagnosticDataElement, DiagnosticParameterElement, DiagnosticParameterIdent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.425: DiagnosticServiceMappingDiagTarget

Class	**DiagnosticServiceSwMapping**			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessedData Prototype	DataPrototype	0..1	iref	This instanceRef identifies the DataPrototype that is supposed to be accessed in the context of the operation argument. **Tags:** atp.Splitkey=accessedDataPrototype, accessed DataPrototype.rootDataPrototypeInCs, accessedData Prototype.contextDataPrototypeInCs, accessedData Prototype.targetDataPrototypeInCs **InstanceRef implemented by:** DataPrototypeInClient ServerInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents a DiagnosticDataElement required to execute the respective diagnostic service in the context of the diagnostic service mapping, **Tags:** atp.Splitkey=diagnosticDataElement
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element. **Tags:** atp.Splitkey=diagnosticParameter
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **Tags:** atp.Splitkey=mappedBswServiceDependency
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy. **Tags:** atp.Splitkey=mappedFlatSwcServiceDependency

△

Class	**DiagnosticServiceSwMapping**			
mappedSwc ServiceDependencyIn System	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition) **Tags:** atp.Splitkey=mappedSwcServiceDependencyIn System, mappedSwcServiceDependencyInSystem.base, mappedSwcServiceDependencyInSystem.contextRoot SwComposition, mappedSwcServiceDependencyIn System.contextSwComponentPrototype, mappedSwc ServiceDependencyInSystem.targetSwcService Dependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element. **Tags:** atp.Splitkey=parameterElementAccess
serviceInstance	DiagnosticService Instance	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping. **Tags:** atp.Splitkey=serviceInstance

Table A.426: DiagnosticServiceSwMapping

Class	**DiagnosticServiceTable**			
Note	This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU. **Tags:** atp.recommendedPackage=DiagnosticServiceTables			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic Connection	DiagnosticConnection	*	ref	This represents the DiagnosticConnection that is taken for handling the data transmission for the enclosing DiagnosticServiceTable. It is possible to refer to more than one diagnostic Connections in order to support more than one diagnostic tester. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticConnection, diagnostic Connection.diagnosticConnection, diagnostic Connection.variationPoint.shortLabel vh.latestBindingTime=postBuild
diagnostic ServiceInstance	DiagnosticService Instance	*	ref	This represents the collection of DiagnosticService Instances to be considered in the scope of this Diagnostic ServiceTable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticServiceInstance, diagnostic ServiceInstance.diagnosticServiceInstance, diagnostic ServiceInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticServiceTable**			
ecuInstance	EcuInstance	0..1	ref	This represents the applicable EcuInstance for this DiagnosticServiceTable. **Tags:** atp.Splitkey=ecuInstance
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.

Table A.427: DiagnosticServiceTable

Class	**DiagnosticSession**			
Note	This meta-class represents the ability to define a diagnostic session. **Tags:** atp.recommendedPackage=DiagnosticSessions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow
jumpToBoot Loader	DiagnosticJumpToBoot LoaderEnum	0..1	attr	This attribute represents the ability to define whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader). If this diagnostic session doesn't allow to jump to Bootloader the value JumpToBootLoaderEnum.noBoot shall be chosen. This Attribute is only used by the AUTOSAR Classic Platform.
p2ServerMax	TimeValue	0..1	attr	This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.
p2StarServer Max	TimeValue	0..1	attr	This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.

Table A.428: DiagnosticSession

Class	**DiagnosticSessionControl**			
Note	This represents an instance of the "Session Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSessionControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic Session	DiagnosticSession	0..1	ref	This represents the applicable DiagnosticSessions **Tags:** atp.Splitkey=diagnosticSession

▽

△

Class	DiagnosticSessionControl			
sessionControl Class	DiagnosticSession ControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSessionControl in the given context. **Tags:** atp.Splitkey=sessionControlClass

Table A.429: DiagnosticSessionControl

Class	DiagnosticSessionControlClass			
Note	This meta-class contains attributes shared by all instances of the "Session Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSessionControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
s3Server Timeout	TimeValue	0..1	attr	Time for the server to keep a diagnostic session other than the default session active while not receiving any diagnostic request message.

Table A.430: DiagnosticSessionControlClass

Class	DiagnosticStartRoutine			
Note	This represents the ability to start a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.start			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table A.431: DiagnosticStartRoutine

Class	DiagnosticStopRoutine			
Note	This represents the ability to stop a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.stop			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table A.432: DiagnosticStopRoutine

Class	DiagnosticStorageCondition			
Note	Specification of a storage condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.433: DiagnosticStorageCondition

Class	DiagnosticStorageConditionGroup			
Note	Storage condition group which includes one or several storage conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticConditionGroup*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
storage Condition	DiagnosticStorage Condition	*	ref	Reference to storageConditions that are part of the StorageConditionGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=storageCondition, storage Condition.diagnosticStorageCondition, storage Condition.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.434: DiagnosticStorageConditionGroup

Class	DiagnosticStorageConditionPortMapping			
Note	Defines to which SWC service ports with DiagnosticStorageConditionNeeds the DiagnosticStorage Condition is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic Storage Condition	DiagnosticStorage Condition	0..1	ref	Reference to the StorageCondition which is mapped to a SWC service port with DiagnosticStorageCondition Needs. **Tags:** atp.Splitkey=diagnosticStorageCondition
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency

▽

△

Class	**DiagnosticStorageConditionPortMapping**			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table A.435: DiagnosticStorageConditionPortMapping

Class	**DiagnosticTestIdentifier**			
Note	This meta-class represents the ability to create a diagnostic test identifier.			
Base	*ARObject*			
Aggregated by	DiagnosticTestResult.testIdentifier			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
id	PositiveInteger	0..1	attr	This represents the numerical id associated with the diagnostic test identifier. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
uasId	PositiveInteger	0..1	attr	This represents the unit and scaling Id of the diagnostic test result. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.436: DiagnosticTestIdentifier

Class	**DiagnosticTestResult**			
Note	This meta-class represents the ability to define diagnostic test results. **Tags:** atp.recommendedPackage=DiagnosticTestResults			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticEvent, diagnosticEvent.diagnostic Event, diagnosticEvent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
monitored Identifier	Diagnostic MeasurementIdentifier	0..1	ref	This attribute represents the related diagnostic monitored identifier. **Tags:** atp.Splitkey=monitoredIdentifier
testIdentifier	DiagnosticTestIdentifier	0..1	aggr	This attribute represents the applicable test identifier. **Tags:** atp.Splitkey=testIdentifier

▽

△

Class	**DiagnosticTestResult**			
updateKind	DiagnosticTestResult UpdateEnum	0..1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.437: DiagnosticTestResult

Class	**DiagnosticTestRoutineIdentifier**			
Note	This represents the test id of the DiagnosticTestIdentifier. **Tags:** atp.recommendedPackage=DiagnosticTestRoutineIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
id	PositiveInteger	0..1	attr	This represents the numerical id of the DiagnosticTest Identifier (see SAE J1979-DA).
requestData Size	PositiveInteger	0..1	attr	This represents the specified data size for the request message. Unit: byte.
responseData Size	PositiveInteger	0..1	attr	This represents the specified data size for the response message. Unit:byte.

Table A.438: DiagnosticTestRoutineIdentifier

Class	**DiagnosticTroubleCodeGroup**			
Note	The diagnostic trouble code group defines the DTCs belonging together and thereby forming a group. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dtc	DiagnosticTroubleCode	*	ref	This represents the collection of DiagnosticTroubleCodes defined by this DiagnosticTroubleCodeGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dtc, dtc.diagnosticTroubleCode, dtc.variation Point.shortLabel vh.latestBindingTime=postBuild
groupNumber	PositiveInteger	0..1	attr	This represents the base number of the DTC group. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.439: DiagnosticTroubleCodeGroup

Class	**DiagnosticTroubleCodeJ1939**
Note	This meta-class represents the ability to model specific trouble-code related properties for J1939. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes This Class is only used by the AUTOSAR Classic Platform.
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	**DiagnosticTroubleCodeJ1939**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the J1939 DTC. **Tags:** atp.Splitkey=dtcProps
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	DiagnosticTroubleCodeJ1939DtcKindEnum	0..1	attr	This attribute further specifies the DTC in terms of its semantics.
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node. **Tags:** atp.Splitkey=node
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN. **Tags:** atp.Splitkey=spn

Table A.440: DiagnosticTroubleCodeJ1939

Enumeration	**DiagnosticTroubleCodeJ1939DtcKindEnum**
Note	This meta-class represents the ability to further specify a J1939 DTC in terms of its semantics. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	DiagnosticTroubleCodeJ1939.kind
Literal	***Description***
serviceOnly	this represents a DTC that is only relevant for service in a garage, reported by e.g. DM53. **Tags:** atp.EnumerationLiteralIndex=0
standard	This represents a non-specific DTC reported by e.g. DM1. **Tags:** atp.EnumerationLiteralIndex=1

Table A.441: DiagnosticTroubleCodeJ1939DtcKindEnum

Class	**DiagnosticTroubleCodeObd**			
Note	This element is used to define OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
considerPtoStatus	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. true: the event is affected by the Dem PTO handling. false: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**DiagnosticTroubleCodeObd**			
eventReadiness Group	EventObdReadiness Group	0..1	aggr	This aggregation allows for the variant definition of the attribute eventObdReadinessGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventReadinessGroup, eventReadiness Group.eventObdReadinessGroup, eventReadiness Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.442: DiagnosticTroubleCodeObd

Class	**DiagnosticTroubleCodeProps**			
Note	This element defines common Dtc properties that can be reused by different DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodePropss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aging	DiagnosticAging	0..1	ref	Reference to an aging algorithm in case that an aging/ unlearning of the event is allowed. **Tags:** atp.Splitkey=aging
diagnostic Memory	DiagnosticMemory Destination	0..1	ref	Reference to the applicable DiagnosticMemory Destination. **Tags:** atp.Splitkey=diagnosticMemory
extendedData Record	DiagnosticExtended DataRecord	*	ref	Defines the links to an extended data class sampler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=extendedDataRecord, extendedData Record.diagnosticExtendedDataRecord, extendedData Record.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrame	DiagnosticFreezeFrame	*	ref	Define the links to a freeze frame class sampler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=freezeFrame, freezeFrame.diagnosticFreeze Frame, freezeFrame.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
immediateNv DataStorage	Boolean	0..1	attr	Change description for Class immediateNvDataStorage in table "Table A.111: DiagnosticTroubleCodeProps": Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering on first occurrence and shutdown. false: immediate non-volatile storage triggering on shutdown.

▽

△

Class	**DiagnosticTroubleCodeProps**			
legislated FreezeFrame ContentUdsObd	DiagnosticDataIdentifierSet	0..1	ref	This reference identifies the layout of legislated freeze frames used for emission related diagnostics over the UDS protocol such as OBDonUDS or WWH-OBD. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=legislatedFreezeFrameContentUdsObd, legislatedFreezeFrameContentUdsObd.diagnosticDataIdentifierSet, legislatedFreezeFrameContentUdsObd.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
maxNumber FreezeFrame Records	PositiveInteger	0..1	attr	This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.
obdRelevance	Boolean	0..1	attr	The attribute controls whether the enclosing DTC is relevant for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
priority	PositiveInteger	0..1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
significance	DiagnosticSignificanceEnum	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.
snapshot RecordContent	DiagnosticDataIdentifierSet	0..1	ref	This represents the freeze frame layout as a set of DIDs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=snapshotRecordContent, snapshotRecordContent.diagnosticDataIdentifierSet, snapshotRecordContent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.443: DiagnosticTroubleCodeProps

Class	**DiagnosticTroubleCodeUds**			
Note	This element is used to describe diagnostic trouble codes (DTCs). **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
considerPto Status	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. true: the event is affected by the Dem PTO handling. false: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticTroubleCodeUds**			
eventReadiness Group	EventObdReadiness Group	0..1	aggr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventReadinessGroup, eventReadiness Group.eventObdReadinessGroup, eventReadiness Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
functionalUnit	PositiveInteger	0..1	attr	This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
obdDtcValue3 Byte	PositiveInteger	0..1	attr	3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
severity	DiagnosticUdsSeverity Enum	0..1	attr	DTC severity according to ISO 14229-1. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
troubleCode Props	DiagnosticTroubleCode Props	0..1	ref	Defined properties associated with the DemDTC. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=troubleCodeProps, troubleCode Props.diagnosticTroubleCodeProps, troubleCode Props.variationPoint.shortLabel vh.latestBindingTime=postBuild
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
wwhObdDtc Class	DiagnosticWwhObdDtc ClassEnum	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.444: DiagnosticTroubleCodeUds

Class	**DiagnosticTroubleCodeUdsToTroubleCodeObdMapping**			
Note	This meta-class represents the ability to associate an OBD trouble code to a UDS trouble code. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
troubleCode Obd	DiagnosticTroubleCode Obd	0..1	ref	This represents the OBD DTC referenced in the mapping between UDS and OBD DTCs. **Tags:** atp.Splitkey=troubleCodeObd

▽

△

Class	**DiagnosticTroubleCodeUdsToTroubleCodeObdMapping**			
troubleCodeUds	DiagnosticTroubleCodeUds	0..1	ref	This represents the UDS DTC referenced in the mapping between UDS and OBD DTCs. **Tags:** atp.Splitkey=troubleCodeUds

Table A.445: DiagnosticTroubleCodeUdsToTroubleCodeObdMapping

Class	**DiagnosticValueNeeds**			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the DCM which are not related to a particular item. In the case of using a sender receiver communicated value, the related value shall be taken via assigned Data in the role "signalBasedDiagnostics". In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal. This Attribute is only used by the AUTOSAR Classic Platform.
diagnosticValue Access	DiagnosticValueAccess Enum	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute controls whether the data can be read and written or whether it is to be handled read-only.
fixedLength	Boolean	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessing StyleEnum	0..1	attr	This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.

Table A.446: DiagnosticValueNeeds

Class	**DiagnosticVerifyCertificateBidirectional**			
Note	This meta-class represents the subfunction to do a bidirectional verification of the certificate. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.447: DiagnosticVerifyCertificateBidirectional

Class	DiagnosticVerifyCertificateUnidirectional			
Note	This meta-class represents the subfunction to do a unidirectional verification of the certificate. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.448: DiagnosticVerifyCertificateUnidirectional

Class	DiagnosticWriteDataByIdentifier			
Note	This represents an instance of the "Write Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
writeClass	DiagnosticWriteDataBy IdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticWriteDataBy Identifier in the given context. **Tags:** atp.Splitkey=writeClass

Table A.449: DiagnosticWriteDataByIdentifier

Class	DltConfig			
Note	This element defines a Dlt configuration for a specific Ecu. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	EcuInstance.dltConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dltEcu	DltEcu	0..1	ref	Reference to the Ecu representation in the Log And Trace Template. **Tags:** atp.Splitkey=dltEcu
dltLogChannel	DltLogChannel	*	aggr	Describes the DltLogChannels that are configured for the log/trace message output **Tags:** atp.Splitkey=dltLogChannel, dltLogChannel.short Name
globalTime Domain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this DltConfig shall be synchronized with **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTime Domain.globalTimeDomain, globalTimeDomain.variation Point.shortLabel vh.latestBindingTime=systemDesignTime
sessionId Support	Boolean	0..1	attr	This attribute defines whether the sessionId is used or not.
timestamp Support	Boolean	0..1	attr	This attribute defines whether a timestamp shall be added to the Dlt messages or not.

Table A.450: DltConfig

Class	**DltLogChannel**			
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). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltConfig.dltLogChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationContext	DltContext	*	ref	Reference to the Swc that produces the log or trace message. Please note that this reference shall not be set in case that the Bsw module produces the associated log or trace messages. **Tags:** atp.Splitkey=applicationContext
defaultTraceState	DltDefaultTraceStateEnum	0..1	attr	This attributes defines the default trace status.
dltMessage	DltMessage	*	ref	Reference to DltMessages that can be transported over the DltLogChannel in the DltPdu. **Tags:** atp.Splitkey=dltMessage
logChannelId	String	0..1	attr	This attribute identifies the Channel for usage within the Log And Trace protocol.
logTraceDefaultLogThreshold	LogTraceDefaultLogLevelEnum	0..1	attr	This attribute allows to set a log level Threshold for Log Level filtering.
nonVerboseMode	Boolean	0..1	attr	This attribute defines whether this channel supports non-Verbose Dlt messages. If disabled only verbose mode messages shall be used.
rxPduTriggering	PduTriggering	0..1	ref	Reference to DltPdu that is received by the DltLog Channel **Tags:** atp.Splitkey=rxPduTriggering
segmentationSupported	Boolean	0..1	attr	If enabled, segmentation will be used if a DLT message is larger than Pdu.length referenced via DltLogChannel.tx PduTriggering.
txPduTriggering	PduTriggering	0..1	ref	Reference to DltPdu that is transmitted by the DltLog Channel. **Tags:** atp.Splitkey=txPduTriggering

Table A.451: DltLogChannel

Class	**DoIpActivationLineNeeds**			
Note	A DoIP entity needs to be informed when an external tester is attached or activated. The DoIpActivationServiceNeeds 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	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.452: DoIpActivationLineNeeds

Class	**DoIpGidNeeds**
Note	The DoIpGidNeeds 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 DoIpGidSynchronizationNeeds and is necessary if the GID can not be provided out of the DoIP configuration options.

▽

△

Class	DoIpGidNeeds			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.453: DoIpGidNeeds

Class	DoIpGidSynchronizationNeeds			
Note	The DoIpGidSynchronizationNeeds 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	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.454: DoIpGidSynchronizationNeeds

Class	DoIpLogicAddress			
Note	The logical DoIP address.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpConfig.logicAddress, DoIpTpConfig.doIpLogicAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
address	Integer	0..1	attr	The logical DoIP address.
doIpLogic AddressProps	AbstractDoIpLogic AddressProps	0..1	aggr	Collection of additional LogicAddress properties. **Tags:** atp.Splitkey=doIpLogicAddressProps, doIpLogic AddressProps.shortName

Table A.455: DoIpLogicAddress

Class	DoIpPowerModeStatusNeeds			
Note	The DoIpPowerModeStatusNeeds indicates that the software-component owning this ServiceNeeds is providing the PowerModeStatus for the DoIP service 0x4003 according to ISO 13400-2:2012.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.456: DoIpPowerModeStatusNeeds

Class	**DoIpServiceNeeds** (abstract)			
Note	This represents an abstract base class for ServiceNeeds related to DoIP.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Subclasses	DoIpActivationLineNeeds, DoIpGidNeeds, DoIpGidSynchronizationNeeds, DoIpPowerModeStatus Needs, DoIpRoutingActivationAuthenticationNeeds, DoIpRoutingActivationConfirmationNeeds, Further ActionByteNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.457: DoIpServiceNeeds

Class	**DoIpTpConfig**			
Note	This element defines exactly one DoIpTp Configuration that is used to configure all DoIPChannels available in a DoIpInterface. Each DoIPChannel describes a connection between a doIpSourceAddress and a doIpTargetAddress and the exchange of DcmIPdus between the PduR and DoIP. **Tags:** atp.recommendedPackage=TpConfigs			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
doIpLogic Address	DoIpLogicAddress	*	aggr	Collection of logical DoIP Addresses. **Tags:** atp.Splitkey=doIpLogicAddress, doIpLogic Address.shortName
tpConnection	DoIpTpConnection	*	aggr	Collection of unidirectional connections between a source address and a target address. **Tags:** atp.Splitkey=tpConnection, tp Connection.ident.shortName

Table A.458: DoIpTpConfig

Class	**DoIpTpConnection**			
Note	A connection identifies the sender and the receiver of this particular communication. The DoIp module routes a tpSdu through this connection.			
Base	*ARObject*, *TpConnection*			
Aggregated by	DoIpTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
doIpSource Address	DoIpLogicAddress	0..1	ref	Reference to the address of the sender of the tpSdu. **Tags:** atp.Splitkey=doIpSourceAddress
doIpTarget Address	DoIpLogicAddress	0..1	ref	Reference to the address of the receiver of the tpSdu. **Tags:** atp.Splitkey=doIpTargetAddress
tpSdu	PduTriggering	0..1	ref	This reference is used to describe the data exchange between DoIp and the PduR. **Tags:** atp.Splitkey=tpSdu

Table A.459: DoIpTpConnection

Class	Documentation			
Note	This meta-class represents the ability to handle a so called standalone documentation. Standalone means, that such a documentation is not embedded in another ARElement or identifiable object. The standalone documentation is an entity of its own which denotes its context by reference to other objects and instances. **Tags:** atp.recommendedPackage=Documentations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
context	DocumentationContext	*	aggr	This is the context of the particular documentation. **Tags:** atp.Splitkey=context, context.shortName
documentation Content	PredefinedChapter	0..1	aggr	This is the content of the documentation related to the specified contexts. **Tags:** atp.Splitkey=documentationContent xml.sequenceOffset=200

Table A.460: Documentation

Class	«atpMixed» **DocumentationBlock**			
Note	This class represents a documentation block. It is made of basic text structure elements which can be displayed in a table cell.			
Base	*ARObject*			
Aggregated by	ApplicabilityInfo.remark, AUTOSAR.introduction, BlueprintGenerator.introduction, *BlueprintPolicy Modifiable*.blueprintDerivationGuide, *DataMapping*.introduction, DefItem.def, *Describable*.introduction, EcucAddInfoParamValue.value, EcuResourceEstimation.introduction, Entry.entryContents, Frame Mapping.introduction, *GeneralAnnotation*.annotationText, *Identifiable*.introduction, IPdu Mapping.introduction, ISignalMapping.introduction, Item.itemContents, LabeledItem.itemContents, Life CycleInfo.remark, *MappingConstraint*.introduction, MsrQueryP2.msrQueryResultP2, Note.noteText, *ScheduleTableEntry*.introduction, *SignalPathConstraint*.introduction, StructuredReq.conflicts, Structured Req.dependencies, StructuredReq.description, StructuredReq.rationale, StructuredReq.remark, StructuredReq.supportingMaterial, StructuredReq.useCase, SwAxisType.swGenericAxisDesc, Topic Content.blockLevelContent, TraceableText.text, VariationPoint.blueprintCondition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MIFigure	0..1	aggr	This represents a figure in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MIFormula	0..1	aggr	This is a formula in the definition block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=60

▽

△

Class	«atpMixed» **DocumentationBlock**			
labeledList	LabeledList	0..1	aggr	This represents a labeled list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=50
list	List	0..1	aggr	This represents numbered or unnumbered list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=30
msrQueryP2	MsrQueryP2	0..1	aggr	This represents automatically contributed contents provided by an msrquery in the context of Documentation Block.
note	Note	0..1	aggr	This represents a note in the text flow. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=80
p	MultiLanguage Paragraph	0..1	aggr	This is one particular paragraph. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=10
structuredReq	StructuredReq	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=100
trace	TraceableText	0..1	aggr	This represents traceable text in the documentation block. This allows to specify requirements/constraints in any documentation block. The kind of the trace is specified in the category. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=90
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=20

Table A.461: DocumentationBlock

Class	**DocumentationContext**			
Note	This class represents the ability to denote a context of a so called standalone documentation. Note that this is an <<atpMixed>>. The contents needs to be considered as ordered.			
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	Documentation.context			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	DocumentationContext			
feature	AtpFeature	0..1	iref	This refers to a particular feature (instance in the M0 model) to which is the context of the documentation. **Tags:** atp.Splitkey=feature, feature.contextElement, feature.target **InstanceRef implemented by:** AnyInstanceRef
identifiable	Identifiable	0..1	ref	This is an identifiable object which is part of the context of the documentation. **Tags:** atp.Splitkey=identifiable

Table A.462: DocumentationContext

Class	DtcStatusChangeNotificationNeeds			
Note	This meta-class represents the needs of a software-component interested to get information regarding any DTC status change.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
notificationTime	DiagnosticClearDtc NotificationEnum	0..1	attr	This attribute determines the time when the notification about the DTC operation shall be executed. This attribute is only relevant for the configuration of the ClearDtc Notification.

Table A.463: DtcStatusChangeNotificationNeeds

Class	DynamicPart			
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*			
Aggregated by	MultiplexedIPdu.dynamicPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicPart Alternative	DynamicPartAlternative	*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. **Tags:** atp.Splitkey=dynamicPartAlternative

Table A.464: DynamicPart

Class	DynamicPartAlternative			
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*			
Aggregated by	DynamicPart.dynamicPartAlternative			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialDynamic Part	Boolean	0..1	attr	Dynamic part that shall be used to initialize this multiplexed IPdu. Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.

▽

△

Class	**DynamicPartAlternative**			
iPdu	ISignalIPdu	0..1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu. **Tags:** atp.Splitkey=iPdu
selectorField Code	Integer	0..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 A.465: DynamicPartAlternative

Class	**E2EProfileCompatibilityProps**			
Note	This meta-class collects settings for configuration of the E2E state machine. **Tags:** atp.recommendedPackage=E2EProfileCompatibilityPropsCollection			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
combinedNo DataInitCount	Boolean	0..1	attr	E2E State machine behavior concerning counting of detected counter errors and missing messages in states NODATA and INIT • value = 0 (false) or not defined: counting of detected counter errors and missing messages in states NODATA and INIT are counted per state separated (Autosar R23-11 or former behavior) • value = 1 (true): counting of detected counter errors and missing messages in states NODATA and INIT are counted in total
transitToInvalid Extended	Boolean	0..1	attr	E2E State machine behavior concerning transition from NODATA/INIT to INVALID value=0 (false): no direct transition from NODATA to INVALID, no transition from INIT to INVALID due to counter-related faults (Autosar R19-11 or former behavior) value=1 (true): direct transition from NODATA to INVALID covered, transition from INIT to INVALID due to counter-related faults covered (state machine extended)

Table A.466: E2EProfileCompatibilityProps

Class	**ECUMapping**			
Note	ECUMapping allows to assign an ECU hardware type (defined in the ECU Resource Template) to an ECUInstance used in a physical topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.ecuResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
commController Mapping	Communication ControllerMapping	*	aggr	The ECUMapping contains the mapping of all CommunicationControllers of the ECU. **Tags:** atp.Splitkey=commControllerMapping

▽

△

Class	**ECUMapping**			
ecu	HwElement	0..1	ref	Reference to a HwElement of category ECU in the ECU Resource Template. **Tags:** atp.Splitkey=ecu
ecuInstance	EcuInstance	0..1	ref	Reference to the EcuInstance in the System Template **Tags:** atp.Splitkey=ecuInstance
hwPortMapping	HwPortMapping	*	aggr	The ECUMapping contains the mapping of all HW Communication Ports of the ECU. **Tags:** atp.Splitkey=hwPortMapping

Table A.467: ECUMapping

Class	**EOCEventRef**			
Note	This is used to define a reference to an `RTEEvent` or `BswEvent`. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EOCExecutableEntityRefAbstract*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ExecutionOrderConstraint.orderedElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModule Instance	BswImplementation	0..1	ref	Specifies the BSW module instance the BSW event is related to. **Tags:** atp.Splitkey=bswModuleInstance
component	SwComponent Prototype	0..1	iref	This association references the specific instance of the SW-C prototype. **Tags:** atp.Splitkey=component, component.context Component, component.targetComponent **InstanceRef implemented by:** ComponentIn CompositionInstanceRef
event	AbstractEvent	0..1	ref	The AbstractEvent (event) whose execution order is restricted by the contraint. **Tags:** atp.Splitkey=event
successor	EOCExecutableEntity RefAbstract	*	ref	The logical successor of an executable entity or a group of executable entities. **Tags:** atp.Splitkey=successor

Table A.468: EOCEventRef

Class	**EOCExecutableEntityRef**			
Note	Defines a reference to a `RunnableEntity` or `BswModuleEntity` If the `ExecutionOrderConstraint` is defined on VFB, System or ECU level, a reference to the `component` the `ExecutableEntity` belongs to shall be provided as context. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EOCExecutableEntityRefAbstract*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ExecutionOrderConstraint.orderedElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModule Instance	BswImplementation	0..1	ref	Specifies the BSW module instance the BSW module entity belongs to. **Tags:** atp.Splitkey=bswModuleInstance

▽

△

Class	EOCExecutableEntityRef			
component	SwComponent Prototype	0..1	iref	This association references the specific instance of the SW-C prototype. **Tags:** atp.Splitkey=component, component.context Component, component.targetComponent **InstanceRef implemented by:** ComponentIn CompositionInstanceRef
executable	ExecutableEntity	0..1	ref	The ExecutableEntity whose execution order is restricted by the contraint. **Tags:** atp.Splitkey=executable
successor	EOCExecutableEntity RefAbstract	*	ref	The logical successor of an executable entity or a group of executable entities. **Tags:** atp.Splitkey=successor

Table A.469: EOCExecutableEntityRef

Class	*EOCExecutableEntityRefAbstract* (abstract)			
Note	Abstract class for referencing atomic `ExecutableEntity`s and/or composites (`EOCExecutableEntityRefGroup`). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	EOCEventRef, EOCExecutableEntityRef, EOCExecutableEntityRefGroup			
Aggregated by	ExecutionOrderConstraint.orderedElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
directSuccessor	EOCExecutableEntity RefAbstract	*	ref	The direct successor of an executable entity or a group of executable entities. **Tags:** atp.Splitkey=directSuccessor

Table A.470: EOCExecutableEntityRefAbstract

Class	EOCExecutableEntityRefGroup			
Note	Specifies a group (composite view) of `ExecutableEntity`s. A `EOCExecutableEntityRefGroup` may reference further `EOCExecutableEntityRefGroup`s. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EOCExecutableEntityRefAbstract*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ExecutionOrderConstraint.orderedElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
letData Exchange Paradigm	LetDataExchange ParadigmEnum	0..1	attr	Specifies the data exchange paradigm between `ExecutableEntity`s within a LET interval.
letInterval	TimingDescriptionEvent Chain	*	ref	This association references the TimingDescriptionEvent Chain that plays the role of a LET interval the executable entities in the group are assigned to. [constr_4554] applies. **Tags:** atp.Splitkey=letInterval
maxCycle Repetitions	PositiveInteger	0..1	attr	**Repetitive Execution Order Constraint only:** The number of repetitions (cycles) of the event in the Repetitive Execution Order Constraint.

▽

△

Class	**EOCExecutableEntityRefGroup**			
maxSlotsPer Cycle	PositiveInteger	0..1	attr	**Repetitive Execution Order Constraint only:** The number of `ExecutableEntity`s (slots) that are executed in a given order within a cycle, for the Repetitive Execution Order Constraint.
nestedElement (ordered)	EOCExecutableEntity RefAbstract	*	ref	This association is used to establish hierarchies of EOCEER Groups and References.
successor	EOCExecutableEntity RefAbstract	*	ref	The logical successor of an executable entity or a group of executable entities. **Tags:** atp.Splitkey=successor
triggeringEvent	TimingDescriptionEvent	0..1	ref	In case of a Repetitive Execution Order Constraint this association references the timing description event triggering every cycle. **Tags:** atp.Splitkey=triggeringEvent

Table A.471: EOCExecutableEntityRefGroup

Class	**EcuAbstractionSwComponentType**			
Note	The ECUAbstraction is a special AtomicSwComponentType that resides between a software-component that wants to access ECU periphery and the Microcontroller Abstraction. The EcuAbstractionSw ComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
hardware Element	HwDescriptionEntity	*	ref	Reference from the EcuAbstractionComponentType to the description of the used HwElements. **Tags:** atp.Splitkey=hardwareElement

Table A.472: EcuAbstractionSwComponentType

Class	**EcuInstance**			
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	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**EcuInstance**			
associatedComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignal IPduGroups 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 ISignalIPdu Groups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup. **Tags:** atp.Splitkey=associatedComIPduGroup This Attribute is only used by the AUTOSAR Classic Platform.
associated Consumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=associatedConsumedProvidedService InstanceGroup, associatedConsumedProvidedService InstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstance Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU. **Tags:** atp.Splitkey=associatedPdurIPduGroup
channel Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.
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. **Tags:** atp.Splitkey=clientIdRange
com Configuration GwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationRx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
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 numberOf Repetitions > 0). This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	EcuInstance			
commController	Communication Controller	*	aggr	CommunicationControllers of the ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commController, commController.shortName, commController.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
dltConfig	DltConfig	0..1	aggr	Describes the Dlt configuration on this EcuInstance. **Tags:** atp.Splitkey=dltConfig This Attribute is only used by the AUTOSAR Classic Platform.
doIpConfig	DoIpConfig	0..1	aggr	DoIp configuration on this EcuInstance. **Tags:** atp.Splitkey=doIpConfig atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ecuTaskProxy	OsTaskProxy	*	ref	Reference to OsTaskProxies assigned to the Ecu Instance. **Tags:** atp.Splitkey=ecuTaskProxy This Attribute is only used by the AUTOSAR Classic Platform.
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. This Attribute is only used by the AUTOSAR Classic Platform.
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of an EcuInstance. **Tags:** atp.Splitkey=firewallRule
j1939Node	J1939Node	*	aggr	Optional collection of J1939Nodes defined on this Ecu Instance. **Tags:** atp.Splitkey=j1939Node, j1939Node.shortName This Attribute is only used by the AUTOSAR Classic Platform.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu. **Tags:** atp.Splitkey=partition, partition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
pncNmRequest	Boolean	0..1	attr	Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
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.

▽

△

Class	**EcuInstance**			
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	0..1	attr	Specifies whether the ECU instance may be put to a "low power mode" • 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. This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpIcmpProps	EthTcpIpIcmpProps	0..1	ref	EcuInstance specific ICMP (Internet Control Message Protocol) attributes **Tags:** atp.Splitkey=tcpIpIcmpProps This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpProps	EthTcpIpProps	0..1	ref	EcuInstance specific TcpIp Stack attributes. **Tags:** atp.Splitkey=tcpIpProps This Attribute is only used by the AUTOSAR Classic Platform.
wakeUpOver BusSupported	Boolean	0..1	attr	Driver support for wakeup over Bus. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.473: EcuInstance

Class	**EcuPartition**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.partition			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
execInUser Mode	Boolean	0..1	attr	A partition can execute either in CPU user mode (execIn UserMode = TRUE) or supervisor mode (execInUser Mode = 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 A.474: EcuPartition

Class	**EcuPartitionToCoreMapping**			
Note	This element maps an EcuPartion to a Core that is represented by a HwElement with category ProcessingUnit. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.ecuPartitionToCoreMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**EcuPartitionToCoreMapping**			
coreId	PositiveInteger	0..1	attr	Optional definition of Core Id.
ecuPartition	EcuPartition	0..1	ref	EcuPartition that is mapped to a ProcessingUnit **Tags:** atp.Splitkey=ecuPartition
processingUnit	HwElement	0..1	ref	HwElement with category ProcessingUnit to which the EcuPartion is mapped. **Tags:** atp.Splitkey=processingUnit

Table A.475: EcuPartitionToCoreMapping

Class	**EcuResourceEstimation**			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	*ARObject*			
Aggregated by	SystemMapping.resourceEstimation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the basic software. **Tags:** atp.Splitkey=bswResourceEstimation, bsw ResourceEstimation.shortName
ecuInstance	EcuInstance	0..1	ref	Reference to the ECU this estimation is done for. **Tags:** atp.Splitkey=ecuInstance
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ecu resource estimation **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-10
rteResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the run time environment. **Tags:** atp.Splitkey=rteResourceEstimation, rteResource Estimation.shortName
swCompToEcu Mapping	SwcToEcuMapping	*	ref	References to SwcToEcuMappings that have been taken into account for the resource estimations. This way it is possible to define dfferent EcuResourceEstimations with diifferent mappings, e.g. before and after mapping an additional SW component. **Tags:** atp.Splitkey=swCompToEcuMapping This Attribute is only used by the AUTOSAR Classic Platform.

Table A.476: EcuResourceEstimation

Class	**EcuTiming**
Note	A timing view used to define timing descriptions and constraints within the scope of one ECU. TimingDescriptions aggregated by EcuTiming are allowed to use all events derived from the class Timing DescriptionEvent. **Tags:** atp.recommendedPackage=TimingExtensions This Class is only used by the AUTOSAR Classic Platform.
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*
Aggregated by	ARPackage.element

▽

△

Class	**EcuTiming**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecu Configuration	EcucValueCollection	0..1	ref	This defines the scope of an EcuTiming. All corresponding timing descriptions and constraints shall be defined within this scope. **Tags:** atp.Splitkey=ecuConfiguration

Table A.477: EcuTiming

Class	***EcucAbstractInternalReferenceDef*** (abstract)			
Note	Common abstract class to gather attributes for internal references (where the destination is located in the Ecu Configuration Description). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractReferenceDef*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	EcucChoiceReferenceDef, EcucReferenceDef, EcucUriReferenceDef			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requires SymbolicName Value	Boolean	0..1	attr	If this attribute is set to true the implementation of the reference is done using a Symbolic Name defined by the referenced container according to TPS_ECUC_02108.

Table A.478: EcucAbstractInternalReferenceDef

Class	***EcucAbstractReferenceDef*** (abstract)			
Note	Common class to gather the attributes for the definition of references. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Subclasses	*EcucAbstractExternalReferenceDef*, *EcucAbstractInternalReferenceDef*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
withAuto	Boolean	0..1	attr	Specifies whether it shall be allowed on the value side to specify this reference value as "AUTO". If withAuto is "true" it shall be possible to set the "isAuto Value" attribute of the respective reference to "true". This means that the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If withAuto is "false" it shall not be possible to set the "is AutoValue" attribute of the respective reference to "true". If withAuto is not present the default is "false".

Table A.479: EcucAbstractReferenceDef

Class	***EcucAbstractReferenceValue*** (abstract)			
Note	Abstract class to be used as common parent for all reference values in the ECU Configuration Description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*			
Subclasses	EcucInstanceReferenceValue, EcucReferenceValue			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
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:** atp.Splitkey=annotation
definition	EcucAbstractReference Def	0..1	ref	Reference to the definition of this EcucAbstractReference Value subclasses in the ECU Configuration Parameter Definition. **Tags:** atp.Splitkey=definition xml.sequenceOffset=-10
isAutoValue	Boolean	0..1	attr	If withAuto is set to "true" for this parameter definition the isAutoValue can be set to "true". If isAutoValue is set to "true" the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If isAutoValue is not present the default is "false".

Table A.480: EcucAbstractReferenceValue

Class	**EcucAddInfoParamValue**			
Note	This parameter corresponds to EcucAddInfoParamDef. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*, *EcucParameterValue*			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	DocumentationBlock	0..1	aggr	Holds the content of the formated text. **Tags:** atp.Splitkey=value

Table A.481: EcucAddInfoParamValue

Class	**EcucChoiceReferenceDef**			
Note	Specify alternative references where in the ECU Configuration description only one of the specified references will actually be used. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractInternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommon Attributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**EcucChoiceReferenceDef**			
destination	EcucContainerDef	*	ref	All the possible parameter containers for the reference are specified. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=destination

Table A.482: EcucChoiceReferenceDef

Class	***EcucCommonAttributes*** (abstract)			
Note	Attributes used by Configuration Parameters as well as References. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucAbstractReferenceDef*, *EcucParameterDef*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
linkTime Support	Boolean	0..1	attr	Indicates whether the parameter or reference can be changed at LinkTime. Please note that Link-Time is only supported in the Vendor Specific Module Definition (VSMD).
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies in which MultiplicityConfigurationClass this parameter or reference is available in a particular ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION, then this aggregation is mandatory. **Tags:** atp.Splitkey=multiplicityConfigClass atp.Status=obsolete xml.namePlural=MULTIPLICITY-CONFIG-CLASSES
origin	String	0..1	attr	String specifying if this configuration parameter is an AUTOSAR standardized configuration parameter or if the parameter is hardware- or vendor-specific.
postBuild Loadable Multiplicity	Boolean	0..1	attr	Indicates whether the multiplicity of the parameter or reference can be changed at post-build loadable time by updating the configuration binary. TRUE means yes, FALSE means no.
postBuild LoadableValue	Boolean	0..1	attr	Indicates whether the parameter or reference value may be loaded to the memory at post-build time . TRUE means yes, FALSE means no.
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a parameter or a reference 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.
postBuildVariant Value	Boolean	0..1	attr	Indicates if a parameter or a reference may have different value 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.

▽

△

Class	***EcucCommonAttributes*** (abstract)			
valueConfig Class	EcucValueConfiguration Class	*	aggr	Specifies in which ValueConfigurationClass this parameter or reference is available in a particular ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION, then this aggregation is mandatory. **Tags:** atp.Splitkey=valueConfigClass atp.Status=obsolete xml.namePlural=VALUE-CONFIG-CLASSES

Table A.483: EcucCommonAttributes

Class	«atpMixedString» **EcucConditionFormula**			
Note	This formula shall yield a boolean expression depending on ecuc queries. Note that the EcucCondition Formula is a mixed string. Therefore, the properties have the upper multiplicity 1. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *FormulaExpression*			
Aggregated by	EcucConditionSpecification.conditionFormula, EcucValidationCondition.validationFormula			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecucQuery	EcucQuery	0..1	ref	The EcucQuery serves as a argument for the formula.
ecucQuery String	EcucQuery	0..1	ref	This indicates that the referenced query shall return a string.

Table A.484: EcucConditionFormula

Enumeration	**EcucConfigurationClassEnum**
Note	Possible configuration classes for the AUTOSAR configuration parameters. **Tags:** atp.Status=obsolete This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*EcucAbstractConfigurationClass*.configClass
Literal	***Description***
Link	Link Time: parts of configuration are delivered from another object code file **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=obsolete
PostBuild	PostBuildTime: after compilation a configuration parameter can be changed. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=obsolete
PreCompile	PreCompile Time: after compilation a configuration parameter can not be changed any more. **Tags:** atp.EnumerationLiteralIndex=2 atp.Status=obsolete

▽

△

Enumeration	**EcucConfigurationClassEnum**
Published Information	PublishedInformation is used to specify the fact that certain information is fixed even before the pre-compile stage. **Tags:** atp.EnumerationLiteralIndex=3 atp.Status=obsolete

Table A.485: EcucConfigurationClassEnum

Class	***EcucContainerDef*** (abstract)			
Note	Base class used to gather common attributes of configuration container definitions. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	EcucChoiceContainerDef, EcucParamConfContainerDef			
Aggregated by	EcucDestinationUriPolicy.container, EcucModuleDef.container, EcucParamConfContainerDef.sub Container			
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 **Tags:** atp.Splitkey=destinationUri
linkTime Support	Boolean	0..1	attr	Indicates whether the container multiplicity can be changed at LinkTime. Please note that Link-Time is only supported in the Vendor Specific Module Definition (VSMD).
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:** atp.Splitkey=multiplicityConfigClass atp.Status=obsolete xml.namePlural=MULTIPLICITY-CONFIG-CLASSES
origin	String	0..1	attr	This attribute specifies whether this configuration container is an AUTOSAR standardized container or whether it is vendor-specific.
postBuild Loadable Multiplicity	Boolean	0..1	attr	Indicates whether the container multiplicity can be changed at post-build loadable time by updating the configuration binary. TRUE means yes, FALSE means no.
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 A.486: EcucContainerDef

Class	**EcucContainerValue**			
Note	Represents a Container definition in the ECU Configuration Description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucContainerValue.subContainer, EcucModuleConfigurationValues.container			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
definition	EcucContainerDef	0..1	ref	Reference to the definition of this Container in the ECU Configuration Parameter Definition. **Tags:** atp.Splitkey=definition xml.sequenceOffset=-10
parameterValue	EcucParameterValue	*	aggr	Aggregates all ECU Configuration Values within this Container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=parameterValue, parameterValue.variation Point.shortLabel vh.latestBindingTime=postBuild
referenceValue	EcucAbstractReference Value	*	aggr	Aggregates all References with this container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=referenceValue, referenceValue.variation Point.shortLabel vh.latestBindingTime=postBuild
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=subContainer, subContainer.shortName, sub Container.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.487: EcucContainerValue

Class	**EcucDefinitionCollection**			
Note	This represents the anchor point of an ECU Configuration Parameter Definition within the AUTOSAR templates structure. **Tags:** atp.recommendedPackage=EcucDefs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
module	EcucModuleDef	*	ref	References to the module definitions of individual software modules. **Tags:** atp.Splitkey=module

Table A.488: EcucDefinitionCollection

Class	***EcucDefinitionElement*** (abstract)
Note	Common class used to express the commonalities of configuration parameters, references and containers. If not stated otherwise the default multiplicity is exactly one mandatory occurrence of the specified element. This Class is only used by the AUTOSAR Classic Platform.

▽

△

Class	***EcucDefinitionElement*** (abstract)			
Base	*ARObject*, *AtpDefinition*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucCommonAttributes*, *EcucContainerDef*, EcucModuleDef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecucCond	EcucCondition Specification	0..1	aggr	If it evaluates to true the Ecu Parameter definition shall be processed as specified. Otherwise the parameter definition shall be ignored. **Tags:** atp.Splitkey=ecucCond xml.sequenceOffset=100
ecucValidation Cond	EcucValidation Condition	*	aggr	Collection of validation conditions which all need to evaluate to true in order to indicate a valid validation condition of the EcucDefinitionElement. **Tags:** atp.Splitkey=ecucValidationCond, ecucValidation Cond.shortName
lowerMultiplicity	PositiveInteger	0..1	attr	The lower multiplicity of the specified element. 0: optional 1: at least one occurrence n: at least n occurrences **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=110
relatedTrace Item	Traceable	0..1	ref	This contains a sloppy reference to the Autosar compatible identifier of the element (EcucId). **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=relatedTraceItem xml.sequenceOffset=-10
scope	EcucScopeEnum	0..1	attr	Specifies the scope of this configuration element. **Tags:** xml.sequenceOffset=150
upperMultiplicity	PositiveInteger	0..1	attr	The upper multiplicity of the specified element. 0: no occurrence (used for VSMD) 1: at most one occurrence m: at most m occurrences If upperMultiplicity is set than upperMultiplicityInfinite shall not be used. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=120
upperMultiplicity Infinite	Boolean	0..1	attr	To express an infinite number of occurrences of this element this attribute has to be set to true. If upperMultiplicityInfinite is set than upperMultiplicity shall not be used. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=130

Table A.489: EcucDefinitionElement

Class	EcucDestinationUriDef			
Note	Description of an EcucDestinationUriDef that is used as target of EcucUriReferenceDefs. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriDefSet.destinationUriDef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
destinationUriPolicy	EcucDestinationUriPolicy	0..1	aggr	Description of the targeted EcucContainerDef. **Tags:** atp.Splitkey=destinationUriPolicy

Table A.490: EcucDestinationUriDef

Class	EcucDestinationUriDefSet			
Note	This class represents a list of EcucDestinationUriDefs. **Tags:** atp.recommendedPackage=EcucDestinationUriDefSets This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
destinationUriDef	EcucDestinationUriDef	*	aggr	This is one particular EcucDestinationUriDef. **Tags:** atp.Splitkey=destinationUriDef, destinationUriDef.shortName

Table A.491: EcucDestinationUriDefSet

Class	EcucDestinationUriPolicy			
Note	The EcucDestinationUriPolicy describes the EcucContainerDef that will be targeted by EcucUriReferenceDefs. The type of the description is dependent of the destinationUriNestingContract attribute. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	EcucDestinationUriDef.destinationUriPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
container	EcucContainerDef	*	aggr	Description of the targetContainer in case that the destinationUriNestingPolicy is set to targetContainer. In all other cases the subContainers of the target container are defined here. **Tags:** atp.Splitkey=container, container.shortName

Table A.492: EcucDestinationUriPolicy

Class	EcucEnumerationLiteralDef			
Note	Configuration parameter type for enumeration literals definition. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucEnumerationParamDef.literal			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**EcucEnumerationLiteralDef**			
ecucCond	EcucCondition Specification	0..1	aggr	If it evaluates to true the literal definition shall be processed as specified. Otherwise the literal definition shall be ignored. **Tags:** atp.Splitkey=ecucCond
origin	String	0..1	attr	String specifying if this literal is an AUTOSAR standardized literal or if the literal is vendor-specific.

Table A.493: EcucEnumerationLiteralDef

Class	**EcucForeignReferenceDef**			
Note	Specify a reference to an XML description of an entity described in another AUTOSAR template. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractExternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
destinationType	String	0..1	attr	The type in the AUTOSAR Metamodel to which instance this reference is allowed to point to.

Table A.494: EcucForeignReferenceDef

Class	**EcucInstanceReferenceDef**			
Note	Specify a reference to an XML description of an entity described in another AUTOSAR template using the INSTANCE REFERENCE semantics. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractExternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
destination Context	String	0..1	attr	The context in the AUTOSAR Metamodel to which' this reference is allowed to point to.
destinationType	String	0..1	attr	The type in the AUTOSAR Metamodel to which' instance this reference is allowed to point to.

Table A.495: EcucInstanceReferenceDef

Class	**EcucInstanceReferenceValue**			
Note	InstanceReference representation in the ECU Configuration. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucAbstractReferenceValue*, *EcucIndexableValue*			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
value	AtpFeature	0..1	iref	InstanceReference representation in the ECU Configuration. **Tags:** atp.Splitkey=value, value.contextElement, value.target **InstanceRef implemented by:** AnyInstanceRef

Table A.496: EcucInstanceReferenceValue

Class	**EcucModuleConfigurationValues**			
Note	Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure. As part of the BSW module description, the EcucModuleConfigurationValues element has two different roles: The recommendedConfiguration contains parameter values recommended by the BSW module vendor. The preconfiguredConfiguration contains values for those parameters which are fixed by the implementation and cannot be changed. These two EcucModuleConfigurationValues are used when the base EcucModuleConfigurationValues (as part of the base ECU configuration) is created to fill parameters with initial values. **Tags:** atp.recommendedPackage=EcucModuleConfigurationValuess This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
container	EcucContainerValue	*	aggr	Aggregates all containers that belong to this module configuration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=container, container.shortName, container.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=10
definition	EcucModuleDef	0..1	ref	Reference to the definition of this EcucModule ConfigurationValues element. Typically, this is a vendor specific module configuration. **Tags:** atp.Splitkey=definition xml.sequenceOffset=-10
ecucDefEdition	RevisionLabelString	0..1	attr	This is the version info of the ModuleDef ECUC Parameter definition to which this values conform to / are based on. For the Definition of ModuleDef ECUC Parameters the AdminData shall be used to express the semantic changes. The compatibility rules between the definition and value revision labels is up to the module's vendor.
implementation ConfigVariant	EcucConfiguration VariantEnum	0..1	attr	Specifies the kind of deliverable this EcucModule ConfigurationValues element provides. If this element is not used in a particular role (e.g. preconfigured Configuration or recommendedConfiguration) then the value shall be one of VariantPreCompile, VariantLink Time, VariantPostBuild. **Tags:** atp.Status=obsolete
module Description	BswImplementation	0..1	ref	Referencing the BSW module description, which this EcucModuleConfigurationValues element is configuring. This is optional because the EcucModuleConfiguration Values element is also used to configure the ECU infrastructure (memory map) or Application SW-Cs. However in case the EcucModuleConfigurationValues are used to configure the module, the reference is mandatory in order to fetch module specific "common" published information. **Tags:** atp.Splitkey=moduleDescription

△

Class	**EcucModuleConfigurationValues**			
postBuildVariant Used	Boolean	0..1	attr	Indicates whether a module implementation has or plans to have (i.e., introduced at link or post-build time) new post-build variation points. TRUE means yes, FALSE means no. If the attribute is not defined, FALSE semantics shall be assumed.

Table A.497: EcucModuleConfigurationValues

Class	**EcucModuleDef**			
Note	Used as the top-level element for configuration definition for Software Modules, including BSW and RTE as well as ECU Infrastructure. **Tags:** atp.recommendedPackage=EcucDefs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpDefinition*, *CollectableElement*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
apiServicePrefix	CIdentifier	0..1	attr	For modules where several instances of the VSMD can be defined the apiServicePrefix defines the API namespace of the derived instances, e.g. Cdd, Xfrm (ComXf, SomeIpXf, E2EXf).
container	EcucContainerDef	*	aggr	Aggregates the top-level container definitions of this specific module definition. **Tags:** atp.Splitkey=container, container.shortName xml.sequenceOffset=11
postBuild Loadable Support	Boolean	0..1	attr	Indicates if the module supports changing the configuration by loading a new configuration binary at postBuild time. TRUE means yes, FALSE means no.
postBuildVariant Support	Boolean	0..1	attr	Indicates if a module supports different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
refinedModule Def	EcucModuleDef	0..1	ref	Optional reference from the Vendor Specific Module Definition to the Standardized Module Definition it refines. In case this EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION this reference shall not be provided. In case this EcucModuleDef has the category VENDOR_SPECIFIC_MODULE_ DEFINITION this reference is mandatory. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=refinedModuleDef
supported ConfigVariant	EcucConfiguration VariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the Ecuc ModuleDef has the category STANDARDIZED_ MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_ MODULE_DEFINITION then this attribute is mandatory. **Tags:** atp.Status=obsolete

Table A.498: EcucModuleDef

Class	**EcucNumericalParamValue**			
Note	Holding the value which is subject to variant handling. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*, *EcucParameterValue*			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	Numerical	0..1	attr	Value which is subject to variant handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.499: EcucNumericalParamValue

Class	**EcucParamConfContainerDef**			
Note	Used to define configuration containers that can hierarchically contain other containers and/or parameter definitions. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucContainerDef*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucChoiceContainerDef.choice, EcucDestinationUriPolicy.container, EcucModuleDef.container, EcucParamConfContainerDef.subContainer			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
parameter	EcucParameterDef	*	aggr	The parameters defined within the EcucParamConfContainerDef. **Tags:** atp.Splitkey=parameter, parameter.shortName
reference	EcucAbstractReferenceDef	*	aggr	The references defined within the EcucParamConfContainerDef. **Tags:** atp.Splitkey=reference, reference.shortName
subContainer	EcucContainerDef	*	aggr	The containers defined within the EcucParamConfContainerDef. **Tags:** atp.Splitkey=subContainer, subContainer.shortName

Table A.500: EcucParamConfContainerDef

Class	***EcucParameterDef*** (abstract)			
Note	Abstract class used to define the similarities of all ECU Configuration Parameter types defined as subclasses. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucAbstractStringParamDef*, EcucAddInfoParamDef, EcucBooleanParamDef, EcucEnumerationParamDef, EcucFloatParamDef, EcucIntegerParamDef			
Aggregated by	EcucParamConfContainerDef.parameter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
derivation	EcucDerivationSpecification	0..1	aggr	A derivation of a Configuration Parameter value can be specified by an informal Calculation Formula or by a formal language that can be used to specify the computational rules. **Tags:** atp.Splitkey=derivation

▽

△

Class	***EcucParameterDef*** (abstract)			
symbolicName Value	Boolean	0..1	attr	Specifies that this parameter's value is used, together with the aggregating container, to derive a symbolic name definition. See chapter "Representation of Symbolic Names" in Ecuc specification for more details.
withAuto	Boolean	0..1	attr	Specifies whether it shall be allowed on the value side to specify this parameter value as "AUTO". If withAuto is "true" it shall be possible to set the "isAuto Value" attribute of the respective parameter to "true". This means that the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If withAuto is "false" it shall not be possible to set the "is AutoValue" attribute of the respective parameter to "true". If withAuto is not present the default is "false".

Table A.501: EcucParameterDef

Class	***EcucParameterValue*** (abstract)			
Note	Common class to all types of configuration values. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*			
Subclasses	EcucAddInfoParamValue, EcucNumericalParamValue, EcucTextualParamValue			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining the ECU Configuration Parameter Values. These are not intended as documentation but are mere design notes. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=10
definition	EcucParameterDef	0..1	ref	Reference to the definition of this EcucParameterValue subclasses in the ECU Configuration Parameter Definition. **Tags:** atp.Splitkey=definition xml.sequenceOffset=-10
isAutoValue	Boolean	0..1	attr	If withAuto is set to "true" for this parameter definition the isAutoValue can be set to "true". If isAutoValue is set to "true" the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If isAutoValue is not present the default is "false". **Tags:** xml.sequenceOffset=20

Table A.502: EcucParameterValue

Class	EcucQuery			
Note	Defines a query to the ECUC Description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucConditionSpecification.ecucQuery, EcucDerivationSpecification.ecucQuery, EcucValidationCondition.ecucQuery			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecucQuery Expression	EcucQueryExpression	0..1	aggr	This is the EcucQuery used in the calculation formula or the condition formula. **Tags:** atp.Splitkey=ecucQueryExpression

Table A.503: EcucQuery

Class	«atpMixedString» **EcucQueryExpression**			
Note	Defines a query expression to the ECUC Description and output the result as an numerical value. Due to the "mixedString" nature of the formula there can be several EcuQueryExpressions used. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	EcucQuery.ecucQueryExpression			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
configElement DefGlobal	EcucDefinitionElement	0..1	ref	The EcucQueryExpression points to an EcucDefinition Element that is used to find an element in the Ecuc Description. In order to find the right element in the Ecuc Description a search is necessary. If the complete Ecuc Description needs to be searched this global reference shall be used. Due to the "mixedString" nature of the EcucQueryExpression several references to Ecuc DefintionElements can be used in one EcucQuery Expression. **Stereotypes:** atpUriDef
configElement DefLocal	EcucDefinitionElement	0..1	ref	The EcucQueryExpression points to an EcucDefinition Element that is used to find an element in the Ecuc Description. In order to find the right element in the Ecuc Description a search is necessary. If the search is executed inside of the same module that contains the EcucQuery this local reference shall be used. Due to the "mixedString" nature of the EcucQueryExpression several references to EcucDefintionElements can be used in one EcucQueryExpression. **Stereotypes:** atpUriDef

Table A.504: EcucQueryExpression

Class	**EcucReferenceDef**			
Note	Specify references within the ECU Configuration Description between parameter containers. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractInternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	EcucReferenceDef			
destination	EcucContainerDef	0..1	ref	Exactly one reference to a parameter container is allowed as destination. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=destination

Table A.505: EcucReferenceDef

Class	EcucReferenceValue			
Note	Used to represent a configuration value that has a parameter definition of type EcucAbstractReference Def (used for all of its specializations excluding EcucInstanceReferenceDef). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucAbstractReferenceValue*, *EcucIndexableValue*			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	Referrable	0..1	ref	Specifies the destination of the reference. **Tags:** atp.Splitkey=value

Table A.506: EcucReferenceValue

Class	EcucTextualParamValue			
Note	Holding a value which is not subject to variation. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*, *EcucParameterValue*			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	VerbatimString	0..1	attr	Value of the parameter, not subject to variant handling.

Table A.507: EcucTextualParamValue

Class	EcucUriReferenceDef			
Note	Definition of reference with a destination that is specified via a destinationUri. With such a reference it is possible to define a reference to a EcucContainerDef in a different module independent from the concrete definition of the target container. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractInternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommon Attributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucParamConfContainerDef.reference			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
destinationUri	EcucDestinationUriDef	0..1	ref	Any EcucContainerDef with a destinationUri that is identical to the destinationUri that is referenced here defines a valid target. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=destinationUri

Table A.508: EcucUriReferenceDef

Class	**EcucValidationCondition**			
Note	Validation condition to perform a formula calculation based on EcucQueries. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*EcucDefinitionElement*.ecucValidationCond			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecucQuery	EcucQuery	*	aggr	Query to the ECU Configuration Description. **Tags:** atp.Splitkey=ecucQuery, ecucQuery.shortName
validation Formula	EcucConditionFormula	0..1	aggr	Definition of the formula used to define validation condition. **Tags:** atp.Splitkey=validationFormula

Table A.509: EcucValidationCondition

Class	**EcucValueCollection**			
Note	This represents the anchor point of the ECU configuration description. **Tags:** atp.recommendedPackage=EcucValueCollections This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecucValue	EcucModule ConfigurationValues	*	ref	References to the configuration of individual software modules that are present on this ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecucValue, ecucValue.ecucModule ConfigurationValues, ecucValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ecuExtract	System	0..1	ref	Represents the extract of the System Configuration that is relevant for the ECU configured with that ECU Configuration Description. **Tags:** atp.Splitkey=ecuExtract

Table A.510: EcucValueCollection

Enumeration	**EndToEndProfileBehaviorEnum**
Note	Behavior of the check functionality
Aggregated by	EndToEndTransformationDescription.profileBehavior
Literal	***Description***
PRE_R4_2	Check has the legacy behavior, before AUTOSAR Release 4.2. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=PRE--R-4--2
R4_2	Check behaves like new P4/P5/P6 profiles introduced in AUTOSAR Release 4.2. **Tags:** atp.EnumerationLiteralIndex=1 xml.name=R-4--2

Table A.511: EndToEndProfileBehaviorEnum

Class	**EndToEndTransformationComSpecProps**			
Note	The class EndToEndTransformationIComSpecProps specifies port specific configuration properties for EndToEnd transformer attributes.			
Base	*ARObject*, *Describable*, *TransformationComSpecProps*			
Aggregated by	ClientComSpec.transformationComSpecProps, *ReceiverComSpec*.transformationComSpecProps, ServerComSpec.transformationComSpecProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearFromValid ToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
disableEndTo EndCheck	Boolean	0..1	attr	Disables/Enables the E2E check. The E2Eheader is removed from the payload independent from the setting of this attribute.
disableEndTo EndState Machine	Boolean	0..1	attr	Disables the E2EStateMachine (only E2E check functionality is performed)
e2eProfile Compatibility Props	E2EProfileCompatibility Props	0..1	ref	Reference to additional settings for the E2E state machine. **Tags:** atp.Splitkey=e2eProfileCompatibilityProps
maxDelta Counter	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.
maxErrorState Init	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.
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. This Attribute is only used by the AUTOSAR Classic Platform.
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. This Attribute is only used by the AUTOSAR Classic Platform.



△

Class	**EndToEndTransformationComSpecProps**			
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 A.512: EndToEndTransformationComSpecProps

Class	**EndToEndTransformationDescription**			
Note	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearFromValid ToInvalid	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.
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.
dataIdNibble Offset	PositiveInteger	0..1	attr	Offset of the Data ID nibble in the Data[] array in bits.
e2eProfile Compatibility Props	E2EProfileCompatibility Props	0..1	ref	Reference to additional settings for the E2E state machine. **Tags:** atp.Splitkey=e2eProfileCompatibilityProps
maxDelta Counter	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.
maxErrorState Init	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.
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.
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.
maxNoNewOr RepeatedData	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.
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.
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.
offset	PositiveInteger	0..1	attr	Offset of the E2E header in the Data[] array in bits.

▽

△

Class	**EndToEndTransformationDescription**			
profileBehavior	EndToEndProfile BehaviorEnum	0..1	attr	Behavior of the check functionality
profileName	NameToken	0..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.
upperHeader BitsToShift	PositiveInteger	0..1	attr	This attribute describes the number of upper-header bits to be shifted. value = 0 or not present: shift of upper header is NOT performed. 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 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 E2 E-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.
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 A.513: EndToEndTransformationDescription

Class	«atpVariation» **EndToEndTransformationISignalProps**			
Note	Holds all the ISignal specific attributes for the EndToEndTransformer.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
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.

▽

△

Class	«atpVariation» **EndToEndTransformationISignalProps**			
sourceId	PositiveInteger	0..1	attr	This attribute represents a unique numerical identifier identifying the source of a certain transmission. In case of C/S communication, this ID uniquely identifies the client. 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.

Table A.514: EndToEndTransformationISignalProps

Class	***EngineeringObject*** (abstract)			
Note	This class specifies an engineering object. Usually such an object is represented by a file artifact. The properties of engineering object are such that the artifact can be found by querying an ASAM catalog file. The engineering object is uniquely identified by domain+category+shortLabel+revisionLabel.			
Base	*ARObject*			
Subclasses	AutosarEngineeringObject, BuildEngineeringObject, Graphic			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	NameToken	0..1	attr	This denotes the role of the engineering object in the development cycle. Categories are such as • SWSRC for source code • SWOBJ for object code • SWHDR for a C-header file Further roles need to be defined via Methodology. **Tags:** xml.sequenceOffset=20
domain	NameToken	0..1	attr	This denotes the domain in which the engineering object is stored. This allows to indicate various segments in the repository keeping the engineering objects. The domain may segregate companies, as well as automotive domains. Details need to be defined by the Methodology. Attribute is optional to support a default domain. **Tags:** xml.sequenceOffset=40
revisionLabel	RevisionLabelString	*	attr	This is a revision label denoting a particular version of the engineering object. **Tags:** xml.sequenceOffset=30
shortLabel	NameToken	0..1	attr	This is the short name of the engineering object. Note that it is modeled as NameToken and not as Identifier since in ASAM-CC it is also a NameToken. **Tags:** xml.sequenceOffset=10

Table A.515: EngineeringObject

Class	**EnumerationMappingTable**
Note	This class represents an attribute value variation point for Enumeration attributes. Note that this class might be used in the extended meta-model only. **Tags:** atp.recommendedPackage=EnumerationMappingTables
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**EnumerationMappingTable**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
entry	EnumerationMapping Entry	*	aggr	Key-value pair mapping enumeration values to unique integers. **Tags:** atp.Splitkey=entry xml.roleElement=true xml.roleWrapperElement=true xml.typeElement=false xml.typeWrapperElement=false

Table A.516: EnumerationMappingTable

Class	**ErrorTracerNeeds**			
Note	Specifies the need to report failures to the error tracer.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tracedFailure	TracedFailure	*	aggr	list of traced failures **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tracedFailure, tracedFailure.shortName, tracedFailure.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.517: ErrorTracerNeeds

Class	**EthGlobalTimeDomainProps**			
Note	Enables the definition of Ethernet Global Time specific properties.			
Base	*ARObject*, *AbstractGlobalTimeDomainProps*			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
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. **Tags:** atp.Splitkey=crcFlags
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	EthGlobalTime ManagedCouplingPort	*	aggr	Collection of CouplingPorts which are managed in the scope of this Ethernet GlobalTimeDomain. **Tags:** atp.Splitkey=managedCouplingPort
message Compliance	EthGlobalTimeMessage FormatEnum	0..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 A.518: EthGlobalTimeDomainProps

Class	**EthGlobalTimeManagedCouplingPort**			
Note	Specifies a CouplingPort which is managed by an Ethernet Global Time Domain.			
Base	*ARObject*			
Aggregated by	EthGlobalTimeDomainProps.managedCouplingPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort	CouplingPort	0..1	ref	Defines which CouplingPort is managed by this EthGlobal TimeManagedCouplingPort. **Tags:** atp.Splitkey=couplingPort
globalTimePort Role	GlobalTimePortRole Enum	0..1	attr	This attribute defines the port behavior.
globalTimeTx Period	TimeValue	0..1	attr	This attribute defines the TX period in seconds
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.
pdelayRespAnd RespFollowUp Timeout	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.
pdelay Response Enabled	Boolean	0..1	attr	Defines whether PDELAY RESPONSE and PDELAY RESPONSE FOLLOW UP shall be sent on this Coupling Port.

Table A.519: EthGlobalTimeManagedCouplingPort

Class	«atpVariation» **EthernetCluster**			
Note	Ethernet-specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort Connection	CouplingPort Connection	*	aggr	Specification of connections between CouplingElements and EcuInstances. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPortConnection, couplingPort Connection.variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingPort StartupActive Time	TimeValue	0..1	attr	The attribute specifies the time in second a coupling port is switched on to enable the host ECU (ECU that maintains an Ethernet switch) to listen to the network for potential network management requests.
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). **Tags:** atp.Splitkey=macMulticastGroup, macMulticast Group.shortName

Table A.520: EthernetCluster

Class	**EthernetCommunicationConnector**			
Note	Ethernet specific attributes to the CommunicationConnector.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ethIpProps	EthIpProps	0..1	ref	EcuInstance specific IP attributes. **Tags:** atp.Splitkey=ethIpProps This Attribute is only used by the AUTOSAR Classic Platform.
maximum Transmission Unit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
neighborCache Size	PositiveInteger	0..1	attr	This attribute specifies the size of neighbor cache or ARP table in units of entries.

Table A.521: EthernetCommunicationConnector

Class	«atpVariation» **EthernetCommunicationController**			
Note	Ethernet specific communication port attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canXlConfig	AbstractCan Communication Controller	0..1	ref	If the Ethernet frames handled by this Ethernet CommunicationController are to be tunneled through CAN XL, then this reference shall refer to the Abstract CanCommunicationController that aggregates the Can ControllerXlConfiguration of the physical CAN XL channel to be used for tunneling. **Tags:** atp.Splitkey=canXlConfig
couplingPort	CouplingPort	*	aggr	Optional CouplingPort that can be used to connect the ECU to a CouplingElement (e.g. a switch). **Tags:** atp.Splitkey=couplingPort, couplingPort.shortName
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the ethernet controller.
macUnicast Address	MacAddressString	0..1	attr	Media Access Control address (MAC address) that uniquely identifies each EthernetCommunication Controller 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.
slaveActAs Passive Communication Slave	Boolean	0..1	attr	This attribute specifies if the EcuInstance is acting as a passive communication slave on the connected Physical Channel. This is used for EthernetCommunication Controllers that use Ethernet hardware which supports wake-up and sleep on the network (e.g. Open Alliance TC10 compliant Ethernet hardware).
slaveQualified UnexpectedLink DownTime	TimeValue	0..1	attr	This attribute specifies time when an unexpected link down is evaluated as link down and indicated to the AUTOSAR communication stack.

Table A.522: EthernetCommunicationController

Enumeration	**EthernetConnectionNegotiationEnum**
Note	Specifies connection negotiation types of Ethernet transceiver links.
Aggregated by	CouplingPort.connectionNegotiationBehavior
Literal	***Description***
auto	Automatic Negotiation **Tags:** atp.EnumerationLiteralIndex=0
master	Master **Tags:** atp.EnumerationLiteralIndex=1
slave	Slave **Tags:** atp.EnumerationLiteralIndex=2

Table A.523: EthernetConnectionNegotiationEnum

Enumeration	**EthernetCouplingPortSchedulerEnum**
Note	Defines the schedule algorithm to be used.
Aggregated by	CouplingPortScheduler.portScheduler
Literal	***Description***
enhancedTraffic Shaper	Scheduler used for enhanced traffic shaping (e.g. weighted round robin) **Tags:** atp.EnumerationLiteralIndex=3
strictPriority	Schedule algorithm "strict priority" **Tags:** atp.EnumerationLiteralIndex=1

Table A.524: EthernetCouplingPortSchedulerEnum

Enumeration	**EthernetMacLayerTypeEnum**
Note	Specifies MAC (Media Access Control) Layer types.
Aggregated by	CouplingPort.macLayerType, EthernetCommunicationController.macLayerType
Literal	***Description***
xGMII	Mac layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII) **Tags:** atp.EnumerationLiteralIndex=1 xml.name=XG-MII
xMII	Mac layer interface (data) bandwith class 100Mbit/s and 10Mbit/s (e.g. RMII, RvMII, SMII, RvMII) **Tags:** atp.EnumerationLiteralIndex=0 xml.name=X-MII
xXGMII	Mac layer interface (data) bandwith class 10Gbit/s **Tags:** atp.EnumerationLiteralIndex=2 xml.name=XXG-MII

Table A.525: EthernetMacLayerTypeEnum

Class	**EthernetPhysicalChannel**			
Note	The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**EthernetPhysicalChannel**			
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_ COMMUNICATION sub-state COMM_FULL_COM_ READY_SLEEP is left in the ComM channel state machine.
network Endpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan. **Tags:** atp.Splitkey=networkEndpoint, network Endpoint.shortName
soAdConfig	SoAdConfig	0..1	aggr	SoAd Configuration for one specific Physical Channel. **Tags:** atp.Splitkey=soAdConfig This Attribute is only used by the AUTOSAR Classic Platform.
vlan	VlanConfig	0..1	aggr	VLAN Configuration. **Tags:** atp.Splitkey=vlan, vlan.shortName

Table A.526: EthernetPhysicalChannel

Enumeration	**EthernetPhysicalLayerTypeEnum**
Note	Specifies physical layer types of Ethernet transceiver links.
Aggregated by	CouplingPort.physicalLayerType
Literal	*Description*
_10000BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 10Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=13 xml.name=10000BASE-T1
_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. 100 BASE-T1 is the IEEE Standardized version of BroadRReach. **Tags:** atp.EnumerationLiteralIndex=7 xml.name=100BASE-T1
_100BASE_TX	Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs. **Tags:** atp.EnumerationLiteralIndex=5 xml.name=100BASE-TX
_10BASE_T1S	Physical layer interface 10BASE-T1S (10Mbit/s, 2 pairs). Used for automotive. **Tags:** atp.EnumerationLiteralIndex=10 atp.Status=draft xml.name=10BASE-T1S
_2500BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 2.5Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=11 xml.name=2500BASE-T1

▽

△

Enumeration	**EthernetPhysicalLayerTypeEnum**
_5000BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 5Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=12 xml.name=5000BASE-T1
iEEE802_11P	Ethernet Standard (IEEE 802.11p) to support wireless communication in vehicular environments. **Tags:** atp.EnumerationLiteralIndex=9 xml.name=IEEE802-11P

Table A.527: EthernetPhysicalLayerTypeEnum

Class	**EthernetPriorityRegeneration**			
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*			
Aggregated by	CouplingPortDetails.ethernetPriorityRegeneration			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ingressPriority	PositiveInteger	0..1	attr	Message priority of the incoming message. range: 0-7
regenerated Priority	PositiveInteger	0..1	attr	Regenerated message priority. range: 0-7

Table A.528: EthernetPriorityRegeneration

Class	**EthernetVlanTranslationTable**			
Note	This element defines one ingress Vlan translation entry in which the IngressVlanID from the incoming frame is replaced by the TranslatedVlanID.			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.vlanTranslationTable			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ingressVlanId	PositiveInteger	0..1	attr	Incoming VlanID from received frame
translatedVlanId	PositiveInteger	0..1	attr	Mapped VlanID after ingress Vlan translation

Table A.529: EthernetVlanTranslationTable

Class	**EthernetWakeupSleepOnDatalineConfig**			
Note	EthernetWakeupSleepOnDatalineConfigSet is the main element that aggregates different config set regarding the wakeup and sleep on data line. An EthernetWakeupSleepOnDatalineConfigSet could aggregate multiple different configurations regarding the wakeup and sleep on dataline (EthernetWakeupSleepOnDatalineConfig).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetWakeupSleepOnDatalineConfigSet.ethernetWakeupSleepOnDatalineConfig			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**EthernetWakeupSleepOnDatalineConfig**			
sleepMode ExecutionDelay	TimeValue	0..1	attr	Delay in seconds to perform a sleep request if the Ethernet hardware (PHY) detect a pending wake-up. This is used to avoid the race condition, if a sleep was requested while a wake-up of a neighboring PHY was received via a local wake-up connection (e.g. I/O pin).
sleepRepetition DelayOfSleep Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a sleep request. This is used to retry a synchronized shutdown of the connected Ethernet hardware (PHY) of the link partner.
sleep RepetitionsOf SleepRequest	PositiveInteger	0..1	attr	Count of repetitions for a sleep on dataline. If a sleep is rejected by the linked communication partner, the sleep is repeated until the count of repetitions exceed. If count of repetitions exceed, the Ethernet hardware (PHY) transit to sleep without acknowledgement of the connected link partner.
wakeupForward LocalEnabled	Boolean	0..1	attr	If enabled, then a remote wake-up received on the physical dataline (e.g. 100BASE-T1) is forwarded as local wake-up (e.g. via an I/O pin). If disabled, then a remote wake-up is not forwarded as local wake-up.
wakeupForward RemoteEnabled	Boolean	0..1	attr	If enabled, then a local wake-up is forwarded to the physical dataline (e.g. 100BASE-T1). If disabled, then a local wake-up is not forwarded to the physical dataline.
wakeupLocal DetectionTime	TimeValue	0..1	attr	Specify the detection time if a local wake-up in seconds is present on the local wake-up connection (e.g. I/O pin). A local wake-up has to be present at least for wakeupLocal DetectionTime to be detected a valid local wake-up.
wakeupLocal DurationTime	TimeValue	0..1	attr	Specify the duration of a local wake-up in seconds to be present on the local wake-up connection (e.g. I/O pin).
wakeupLocal Enabled	Boolean	0..1	attr	If enabled, then a local wake-up received via a local connection (e.g. I/O pin) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reacting on a local wake-up.
wakeupRemote Enabled	Boolean	0..1	attr	If enabled, then a remote wake-up received via the physical dataline (e.g. 100BASE-T1) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reaction on a remote wake-up.
wakeup RepetitionDelay OfWakeup Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.
wakeup RepetitionsOf Wakeup Request	PositiveInteger	0..1	attr	Count of repetitions for a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.

Table A.530: EthernetWakeupSleepOnDatalineConfig

Class	**EvaluatedVariantSet**			
Note	This meta class represents the ability to express if a set of ARElements is able to support one or more particular variants. In other words, for a given set of evaluatedElements this meta class represents a table of evaluated variants, where each PredefinedVariant represents one column. In this column each descendant sw SystemconstantValue resp. postbuildVariantCriterionValue represents one entry. In a graphical representation each swSystemconstantValueSet / postBuildVariantCriterionValueSet could be used as an intermediate headline in the table column. If the approvalStatus is "APPROVED" it expresses that the collection of CollectableElements is known be valid for the given evaluatedVariants. Note that the EvaluatedVariantSet is a CollectableElement. This allows to establish a hierarchy of EvaluatedVariantSets. **Tags:** atp.recommendedPackage=EvaluatedVariantSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
approvalStatus	NameToken	0..1	attr	Defines the approval status of a predefined variant. Two values are predefined: "APPROVED" and "REJECTED": • Approved variants are known to work. • Rejected variants are known NOT to work. Further values can be approved on a per-company basis; within AUTOSAR only "APPROVED" and "REJECTED" should be recognized.
evaluated Element	CollectableElement	*	ref	This represents a particular element which is evaluated in context of the EvaluatedVariants. The approvalStatus applies to this element (and all of its descendants). In other words, the referenced elements are those that were considered when the predefined variant was evaluated. **Tags:** atp.Splitkey=evaluatedElement
evaluated Variant	PredefinedVariant	*	ref	This metaclass represents one particular variant which was evaluated. LowerMultiplicity is set to 0 to support a stepwise approach. **Tags:** atp.Splitkey=evaluatedVariant

Table A.531: EvaluatedVariantSet

Class	**EventControlledTiming**			
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*, *Describable*			
Aggregated by	TransmissionModeTiming.eventControlledTiming			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf Repetitions	Integer	0..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. **Tags:** atp.Splitkey=repetitionPeriod

Table A.532: EventControlledTiming

Enumeration	**EventGroupControlTypeEnum**
Note	Types of a RoutingGroups for the event communication.
Aggregated by	PduActivationRoutingGroup.eventGroupControlType
Literal	***Description***
activationAnd TriggerUnicast	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
activationUnicast AndMulticast	Activate the data path for unicast and multicast events of an EventGroup if they use the identical data path. **Tags:** atp.EnumerationLiteralIndex=4
triggerUnicast	Activate the data path for triggered unicast events that are sent out after a client got subscribed. **Tags:** atp.EnumerationLiteralIndex=3

Table A.533: EventGroupControlTypeEnum

Class	**EventHandler**			
Note	This element represents an event group as part of the Provided Service Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ProvidedServiceInstance.eventHandler			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventGroup Identifier	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.
eventMulticast Address	ApplicationEndpoint	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 mulicastThreshold is exceeded. This address is transmitted in the SD-SubscribeEvent GroupAck Message to client (answer to SD-Subscribe EventGroup). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastAddress, eventMulticast Address.applicationEndpoint, eventMulticast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
multicast Threshold	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.
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events. **Tags:** atp.Splitkey=pduActivationRoutingGroup, pdu ActivationRoutingGroup.shortName

△

Class	**EventHandler**			
sdServerEg TimingConfig	SomeipSdServerEvent GroupTimingConfig	0..1	ref	Server Timing configuration settings that are EventGroup specific. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdServerEgTimingConfig, sdServerEg TimingConfig.someipSdServerEventGroupTimingConfig, sdServerEgTimingConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.534: EventHandler

Class	**EventObdReadinessGroup**			
Note	This meta-class represents the ability to define the value of attribute eventObdReadinessGroup. It is only introduced to allow for a variant modeling of this attribute.			
Base	*ARObject*			
Aggregated by	DiagnosticTroubleCodeObd.eventReadinessGroup, DiagnosticTroubleCodeUds.eventReadinessGroup			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
eventObd Readiness Group	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **Stereotypes:** atpIdentityContributor

Table A.535: EventObdReadinessGroup

Class	***EventTriggeringConstraint*** (abstract)			
Note	Describes the occurrence behavior of the referenced timing event. The occurrence behavior can only be determined when a mapping from the timing events to the implementation can be obtained. However, such an occurrence behavior can also be described by the modeler as an assumption or as a requirement about the occurrence of the event.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Subclasses	ArbitraryEventTriggering, BurstPatternEventTriggering, ConcretePatternEventTriggering, PeriodicEvent Triggering, SporadicEventTriggering			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
event	TimingDescriptionEvent	0..1	ref	The referenced timing event **Tags:** atp.Splitkey=event

Table A.536: EventTriggeringConstraint

Class	**ExclusiveArea**			
Note	Prevents an executable entity running in the area from being preempted.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*InternalBehavior*.exclusiveArea			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.537: ExclusiveArea

Class	***ExecutableEntity*** (abstract)			
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 `activationReason` 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`. **Tags:** atp.Splitkey=activationReason, activation Reason.shortName
canEnter	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canEnter, canEnter.exclusiveArea, can Enter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity. **Tags:** atp.Splitkey=exclusiveAreaNestingOrder
minimumStart Interval	TimeValue	0..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 ReentrancyLevel Enum for details. Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.
runsInside	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=runsInside, runsInside.exclusiveArea, runs Inside.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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. **Tags:** atp.Splitkey=swAddrMethod

Table A.538: ExecutableEntity

Class	**ExecutableEntityActivationReason**
Note	This meta-class represents the ability to define the reason for the activation of the enclosing `ExecutableEntity`.
Base	*ARObject*, *ImplementationProps*, *Referrable*
Aggregated by	*ExecutableEntity*.activationReason

▽

△

Class	**ExecutableEntityActivationReason**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bitPosition	PositiveInteger	0..1	attr	This attribute allows for defining the position of the enclosing ExecutableEntityActivationReason in the activation vector.

Table A.539: ExecutableEntityActivationReason

Class	**ExecutionOrderConstraint**			
Note	This constraint is used to restrict the order of execution for a set of `ExecutableEntity`s. The `ExecutionOrderConstraint` can be used in ANY timing view. The various scopes for `ExecutionOrderConstraint` are described below. Generally, each `ExecutionOrderConstraint` has a scope of software components and can reference all `ExecutableEntity`s available in the corresponding internal behavior (`RunnableEntity` and `BswModuleEntity`) either directly or by the events activating respectively starting them: `RTEEvent` and `BswEvent`. • On VFB level an `ExecutionOrderConstraint` can be specified for `RunnableEntity`s - part of the composition hierarchy referenced by the `VfbTiming`. • On SW-C level an `ExecutionOrderConstraint` can be specified for `RunnableEntity`s - part of the `InternalBehavior` referenced by the `SwcTiming`. • On System level an `ExecutionOrderConstraint` can be specified for `RunnableEntity`s - part of the composition hierarchy of the system referenced by the `SystemTiming`. • On BSW module level, an `ExecutionOrderConstraint` can be specified for `BswModuleEntity`s part of an `BswInternalBehavior` referenced by the `BswModuleTiming`. • On ECU level an `ExecutionOrderConstraint` can be specified for all `ExecutableEntity`s and Events available via the EcucValueCollection, covering ECU Extract and BSW Module Configuration, referenced by the `EcuTiming`. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
executionOrder ConstraintType	ExecutionOrder ConstraintTypeEnum	0..1	attr	Specifies the specific type of ExecutionOrderConstraint.
orderedElement	EOCExecutableEntity RefAbstract	*	aggr	This aggregation represents an unordered collection of references to RunnableEntities which shall be considered in the ExecutionOrderConstraint. The role does not imply that the collection of references itself shall be ordered. **Tags:** atp.Splitkey=orderedElement, ordered Element.shortName

Table A.540: ExecutionOrderConstraint

Enumeration	**ExecutionOrderConstraintTypeEnum**
Note	Specifies the type of the `executionOrderConstraintType` for a `ExecutionOrderConstraint`. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	ExecutionOrderConstraint.executionOrderConstraintType
Literal	***Description***
hierarchicalEOC	Specifies that the Execution Order Constraint specifies a hierarchical execution order constraint. **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**ExecutionOrderConstraintTypeEnum**
ordinaryEOC	Specifies that the Execution Order Constraint specifies an ordinary execution order constraint. **Tags:** atp.EnumerationLiteralIndex=1
repetitiveEOC	Specifies that the Execution Order Constraint specifies a repetitive execution order constraint. **Tags:** atp.EnumerationLiteralIndex=2

Table A.541: ExecutionOrderConstraintTypeEnum

Class	***ExecutionTime*** (abstract)			
Note	Base class for several means how to describe the ExecutionTime of software. The required context information is provided through this class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AnalyzedExecutionTime, MeasuredExecutionTime, RoughEstimateOfExecutionTime, Simulated ExecutionTime			
Aggregated by	ResourceConsumption.executionTime			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
exclusiveArea	ExclusiveArea	0..1	ref	Reference to the ExclusiveArea this execution time is provided for. **Tags:** atp.Splitkey=exclusiveArea
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this execution time is described. **Tags:** atp.Splitkey=executableEntity
hardware Configuration	HardwareConfiguration	0..1	aggr	Provides information on the HardwareConfiguration used to specify this ExecutionTime. **Tags:** atp.Splitkey=hardwareConfiguration
hwElement	HwElement	0..1	ref	The hardware element (e.g. type of ECU) for which the execution time is specified. **Tags:** atp.Splitkey=hwElement
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. **Tags:** atp.Splitkey=includedLibrary
memorySection Location	MemorySectionLocation	*	aggr	Provides information on the MemorySectionLocation which is involved in the ExecutionTime description. **Tags:** atp.Splitkey=memorySectionLocation
softwareContext	SoftwareContext	0..1	aggr	Provides information on the detailed SoftwareContext used to provide the ExecutionTime description. **Tags:** atp.Splitkey=softwareContext

Table A.542: ExecutionTime

Class	**ExecutionTimeConstraint**
Note	Constrains the execution time of the referenced `executable` in `component` between a `minimum` and `maximum` interval. The time to execute the `executable` including interruptions by other entities and including external calls is commonly called "response time". The TimingExtensions provide the concept of event chains and latency constraints for that purpose. An event chain from the start of the entity to the termination of the entity with according latency constraint represents a response time constraint for that executable entity. This Class is only used by the AUTOSAR Classic Platform.

▽

△

Class	ExecutionTimeConstraint			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
component	SwComponent Prototype	0..1	iref	The component that containts the referenced Executable Entity for the ExecutionTimeConstraint. If the entity is in a basic software module no component shall be provided. **Tags:** atp.Splitkey=component, component.context Component, component.targetComponent **InstanceRef implemented by:** ComponentIn CompositionInstanceRef
executable	ExecutableEntity	0..1	ref	The referenced ExecutableEntity for the ExecutionTime Constraint. **Tags:** atp.Splitkey=executable
executionTime Type	ExecutionTimeType Enum	0..1	attr	Specifies the type of the execution time constrained by ExecutionTimeConstraint,
maximum	MultidimensionalTime	0..1	aggr	The maximum execution time. **Tags:** atp.Splitkey=maximum
minimum	MultidimensionalTime	0..1	aggr	The minimum execution time. **Tags:** atp.Splitkey=minimum

Table A.543: ExecutionTimeConstraint

Class	ExternalTriggerOccurredEvent			
Note	This event is raised when the referenced Trigger has occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
trigger	Trigger	0..1	iref	The referenced Trigger raises this ExternalTrigger OccurredEvent. **Tags:** atp.Splitkey=trigger, trigger.contextRPort, trigger.targetTrigger **InstanceRef implemented by:** RTriggerInAtomicSwc InstanceRef

Table A.544: ExternalTriggerOccurredEvent

Class	ExternalTriggeringPoint			
Note	If a RunnableEntity owns an ExternalTriggeringPoint it is entitled to raise an ExternalTriggerOccurred Event.			
Base	*ARObject*			
Aggregated by	RunnableEntity.externalTriggeringPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**ExternalTriggeringPoint**			
ident	ExternalTriggeringPoint Ident	0..1	aggr	The aggregation in the role ident provides the ability to make the ExternalTriggeringPoint identifiable. From the semantical point of view, the ExternalTriggering Point is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=-100
trigger	Trigger	0..1	iref	The trigger taken for the ExternalTriggeringPoint. **Tags:** atp.Splitkey=trigger, trigger.contextPPort, trigger.target Trigger xml.namePlural=TRIGGER-IREF xml.roleElement=false xml.roleWrapperElement=true xml.typeElement=true xml.typeWrapperElement=false **InstanceRef implemented by:** PTriggerInAtomicSwc TypeInstanceRef

Table A.545: ExternalTriggeringPoint

Class	***FibexElement*** (abstract)			
Note	ASAM FIBEX elements specifying Communication and Topology.			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*BusMirrorChannelMapping*, *CommunicationCluster*, ConsumedProvidedServiceInstanceGroup, CouplingElement, EcuInstance, EthernetWakeupSleepOnDatalineConfigSet, *Frame*, Gateway, Global TimeDomain, ISignal, ISignalGroup, ISignalIPduGroup, NmConfig, *Pdu*, PdurIPduGroup, Secure CommunicationPropsSet, ServiceInstanceCollectionSet, SocketConnectionIpduIdentifierSet, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.546: FibexElement

Class	**FlatInstanceDescriptor**			
Note	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. Use cases: • 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 Note that in addition it is possible to assign alias names via AliasNameAssignment. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlatMap.instance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**FlatInstanceDescriptor**			
bsw Implementation	BswImplementation	0..1	ref	Reference to BswImplementation that defines the context for the AutosarDataPrototype that is referenced by Flat InstanceDescriptor.dataPrototype. **Tags:** atp.Splitkey=bswImplementation
dataPrototype	AutosarDataPrototype	0..1	ref	Reference to a DataPrototype that is defined in the Bsw InternalBehavior in the context of a BswImplementation that is defined by the FlatInstanceDescriptor.bsw Implementation reference. **Tags:** atp.Splitkey=dataPrototype
ecuExtract Reference	AtpFeature	0..1	iref	Refers to the instance in the ECU extract. This is valid only, if the FlatMap is used in the context of an ECU extract. 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 instance of the component prototype and the Atomic SoftwareComponentType, which is refered by the particular SwcInternalBehavior. **Tags:** atp.Splitkey=ecuExtractReference, ecuExtract Reference.contextElement, ecuExtractReference.target xml.sequenceOffset=40 **InstanceRef implemented by:** AnyInstanceRef
role	Identifier	0..1	attr	The role denotes the particular role of the downstream memory location described by this FlatInstanceDescriptor. 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.
rtePluginProps	RtePluginProps	0..1	aggr	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in. **Tags:** atp.Splitkey=rtePluginProps
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this FlatInstanceDescriptor. **Tags:** atp.Splitkey=swDataDefProps
upstream Reference	AtpFeature	0..1	iref	Refers to the instance in the context of an "upstream" description, which could be: the SYSTEM_ DESCRIPTION, or SYSTEM_EXTRACT, or ECU_ SYSTEM_DESCRIPTION, or SW_CLUSTER_SYSTEM_ DESCRIPTION, or the basic software module description (in this case only the target reference of the AnyInstance Ref is needed), 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. 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 Swc Internal 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. ▽

▽

△

Class	FlatInstanceDescriptor			
				△ **Tags:** atp.Splitkey=upstreamReference, upstream Reference.contextElement, upstreamReference.target xml.sequenceOffset=20 **InstanceRef implemented by:** AnyInstanceRef

Table A.547: FlatInstanceDescriptor

Class	FlatMap			
Note	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. 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. **Tags:** atp.recommendedPackage=FlatMaps This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
instance	FlatInstanceDescriptor	*	aggr	A descriptor instance aggregated in the flat map. 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. The aggregation has been made splitable 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instance, instance.shortName, instance.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.548: FlatMap

Class	FlexrayAbsolutelyScheduledTiming			
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*			
Aggregated by	FlexrayFrameTriggering.absolutelyScheduledTiming			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Cycle	CommunicationCycle	0..1	aggr	The communication cycle where the frame is sent. **Tags:** atp.Splitkey=communicationCycle

▽

△

Class	**FlexrayAbsolutelyScheduledTiming**			
slotID	PositiveInteger	0..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 A.549: FlexrayAbsolutelyScheduledTiming

Enumeration	**FlexrayChannelName**
Note	Name of the channel.
Aggregated by	FlexrayPhysicalChannel.channelName
Literal	***Description***
channelA	Channel A **Tags:** atp.EnumerationLiteralIndex=0
channelB	Channel B **Tags:** atp.EnumerationLiteralIndex=1

Table A.550: FlexrayChannelName

Class	«atpVariation» **FlexrayCluster**			
Note	FlexRay specific attributes to the physicalCluster **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
actionPoint Offset	Integer	0..1	attr	The offset of the action point in networks
bit	TimeValue	0..1	attr	Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPer Bit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLowMax	Integer	0..1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStart Attempts	Integer	0..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	0..1	attr	Length of the cycle. Unit: seconds
cycleCountMax	Integer	0..1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
detectNitError	Boolean	0..1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSlotIdle Phase	Integer	0..1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfterTx	Integer	0..1	attr	Duration for which the bitstrobing is paused after transmission [gdBit].

▽

△

Class	«atpVariation» **FlexrayCluster**			
listenNoise	Integer	0..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	0..1	attr	The number of macroticks in a communication cycle
macrotick Duration	TimeValue	0..1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithout ClockCorrection Fatal	Integer	0..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.
maxWithout ClockCorrection Passive	Integer	0..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.
minislotAction PointOffset	Integer	0..1	attr	The Offset of the action point within a minislot. Unit: macroticks
minislotDuration	Integer	0..1	attr	The duration of a minislot (dynamic segment). Unit: macroticks.
networkIdle Time	Integer	0..1	attr	The duration of the network idle time in macroticks
network Management VectorLength	Integer	0..1	attr	Length of the Network Management vector in a cluster [bytes]
numberOf Minislots	Integer	0..1	attr	Number of Minislots in the dynamic segment.
numberOfStatic Slots	Integer	0..1	attr	The number of static slots in the static segment.
offsetCorrection Start	Integer	0..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
payloadLength Static	Integer	0..1	attr	Globally configured payload length of a static frame. Unit: 16-bit WORDS.
safetyMargin	Integer	0..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 be resynchronized.
sampleClock Period	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlot Duration	Integer	0..1	attr	The duration of a slot in the static segment. Unit: macroticks
symbolWindow	Integer	0..1	attr	The duration of the symbol window. Unit: macroticks
symbolWindow ActionPoint Offset	Integer	0..1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameId CountMax	Integer	0..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.
tranceiver StandbyDelay	Float	0..1	attr	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full Flex Ray 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.

▽

△

Class	«atpVariation» **FlexrayCluster**			
transmission StartSequence Duration	Integer	0..1	attr	Number of bits in the Transmission Start Sequence [gd Bits].
wakeupRxIdle	Integer	0..1	attr	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit:bit Duration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.
wakeupRxLow	Integer	0..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.
wakeupRx Window	Integer	0..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	0..1	attr	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
wakeupTxIdle	Integer	0..1	attr	Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gDbit

Table A.551: FlexrayCluster

Class	**FlexrayCommunicationConnector**			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmReadySleep Time	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.
wakeUp Channel	Boolean	0..1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table A.552: FlexrayCommunicationConnector

Class	**FlexrayFifoConfiguration**			
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*			
Aggregated by	FlexrayCommunicationController.flexrayFifo			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
admitWithout MessageId	Boolean	0..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	0..1	attr	FIFO cycle counter acceptance criteria.
channel	FlexrayPhysicalChannel	0..1	ref	Fifo channel admittance criteria. **Tags:** atp.Splitkey=channel
cycleRepetition	Integer	0..1	attr	FIFO cycle counter acceptance criteria.

▽

△

Class	**FlexrayFifoConfiguration**			
fifoDepth	Integer	0..1	attr	FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.
fifoRange	FlexrayFifoRange	*	aggr	FIFO Frame Id range acceptance criteria. **Tags:** atp.Splitkey=fifoRange
msgIdMask	Integer	0..1	attr	FIFO message identifier acceptance criteria (Mask filter).
msgIdMatch	Integer	0..1	attr	FIFO message identifier acceptance criteria (Match filter).

Table A.553: FlexrayFifoConfiguration

Class	**FlexrayFifoRange**			
Note	FIFO Frame Id range acceptance criteria.			
Base	*ARObject*			
Aggregated by	FlexrayFifoConfiguration.fifoRange			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
rangeMax	Integer	0..1	attr	Max Range.
rangeMin	Integer	0..1	attr	Min Range.

Table A.554: FlexrayFifoRange

Class	**FlexrayFrame**			
Note	FlexRay specific Frame element. **Tags:** atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.555: FlexrayFrame

Class	**FlexrayFrameTriggering**			
Note	FlexRay specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
absolutely Scheduled Timing	FlexrayAbsolutely ScheduledTiming	*	aggr	Specification of a sending behaviour where the exact time for the frames transmission is guaranteed. **Tags:** atp.Splitkey=absolutelyScheduledTiming
allowDynamic LSduLength	Boolean	0..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.

▽

△

Class	**FlexrayFrameTriggering**			
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	0..1	attr	Switching the Payload Preamble bit.

Table A.556: FlexrayFrameTriggering

Class	**FlexrayNmCluster**			
Note	FlexRay specific NM cluster attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterEnabled	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 nmCarWakeUpFilter NodeId is considered as CarWakeUp request.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If Car WakeUp filtering is supported (nmCarWakeUpFilter Enabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUp RxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWake Up bit evaluation in received NmPdus.
nmDataCycle	Integer	0..1	attr	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.
nmMain FunctionPeriod	TimeValue	0..1	attr	Defines the processing cycle of the main function of FrNm module.
nmRemote SleepIndication Time	TimeValue	0..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.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmRepetition Cycle	Integer	0..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	0..1	attr	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.

Table A.557: FlexrayNmCluster

Class	**FlexrayNmClusterCoupling**
Note	FlexRay attributes that are valid for each of the referenced (coupled) FlexRay clusters. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *NmClusterCoupling*

▽

△

Class	FlexrayNmClusterCoupling			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	Type	Mult.	Kind	Note
coupledCluster	FlexrayNmCluster	*	ref	Reference to coupled FlexRay Clusters. **Tags:** atp.Splitkey=coupledCluster
nmSchedule Variant	FlexrayNmSchedule Variant	0..1	attr	FrNm schedule variant according to FrNm SWS.

Table A.558: FlexrayNmClusterCoupling

Class	FlexrayNmNode			
Note	FlexRay specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.559: FlexrayNmNode

Class	FlexrayPhysicalChannel			
Note	FlexRay specific attributes to the physicalChannel			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	Type	Mult.	Kind	Note
channelName	FlexrayChannelName	0..1	attr	Name of the channel (Channel A or Channel B).

Table A.560: FlexrayPhysicalChannel

Class	FlexrayTpConfig			
Note	This element defines exactly one FlexRay ISO TP Configuration. One FlexRayTpConfig element shall be created for each FlexRay Network in the System that uses Flex Ray Iso Tp. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
pduPool	FlexrayTpPduPool	*	aggr	Configuration of FlexRay TP Pdu Pools. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduPool, pduPool.shortName, pdu Pool.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**FlexrayTpConfig**			
tpAddress	TpAddress	*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	FlexrayTpConnection	*	aggr	Configuration of FlexRay TP Connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection Control	FlexrayTpConnection Control	*	aggr	Configuration of FlexRay TP Connection Controls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnectionControl, tpConnection Control.shortName, tpConnectionControl.variation Point.shortLabel vh.latestBindingTime=postBuild
tpEcu	FlexrayTpEcu	*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpEcu, tpEcu.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	FlexrayTpNode	*	aggr	Senders and receivers of FlexRay TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.561: FlexrayTpConfig

Class	**FlexrayTpConnection**			
Note	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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	FlexrayTpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bandwidth Limitation	Boolean	0..1	attr	Specifies whether the connection requires a bandwidth limitation or not.

▽

△

Class	**FlexrayTpConnection**			
directTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=directTpSdu
multicast	TpAddress	0..1	ref	TP address for 1:n connections. **Tags:** atp.Splitkey=multicast
receiver	FlexrayTpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver
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. **Tags:** atp.Splitkey=reversedTpSdu
rxPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpRx PduPool) which are defined for receiving data via this particular connection. 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. **Tags:** atp.Splitkey=rxPduPool
tpConnection Control	FlexrayTpConnection Control	0..1	ref	Reference to the connection control. **Tags:** atp.Splitkey=tpConnectionControl
transmitter	FlexrayTpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter
txPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpTx PduPool) which are defined for sending data via this particular connection. 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. **Tags:** atp.Splitkey=txPduPool

Table A.562: FlexrayTpConnection

Class	**FlexrayTpConnectionControl**			
Note	Configuration parameters to control a FlexRay TP connection. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpConnectionControl			
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".
maxNumberOf NpduPerCycle	Integer	0..1	attr	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.

▽

△

Class	**FlexrayTpConnectionControl**			
maxRetries	Integer	0..1	attr	This parameter defines the maximum number of retries (if retry is configured for the particular channel).
separationCycle Exponent	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 A.563: FlexrayTpConnectionControl

Class	**FlexrayTpEcu**			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	FlexrayTpConfig.tpEcu			
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	0..1	ref	Connection to the ECUInstance in the Topology **Tags:** atp.Splitkey=ecuInstance
fullDuplex Enabled	Boolean	0..1	attr	The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.

Table A.564: FlexrayTpEcu

Class	**FlexrayTpNode**			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpNode			
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). **Tags:** atp.Splitkey=connector
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). **Tags:** atp.Splitkey=tpAddress

Table A.565: FlexrayTpNode

Class	**FlexrayTpPduPool**			
Note	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.pduPool			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nPdu	NPdu	*	ref	Reference to NPdus that are part of the PduPool. **Tags:** atp.Splitkey=nPdu This Attribute is only used by the AUTOSAR Classic Platform.

Table A.566: FlexrayTpPduPool

Class	**ForbiddenSignalPath**			
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*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	SwcToSwcOperation Arguments	*	aggr	Reference to the operation arguments of one operation which shall not take the predefined way in the topology. **Tags:** atp.Splitkey=operation
physical Channel	PhysicalChannel	*	ref	The SwcToSwcSignal shall not be transmitted on one of these physical channels. **Tags:** atp.Splitkey=physicalChannel
signal	SwcToSwcSignal	*	aggr	The data element which shall not take the predefined way in the topology. **Tags:** atp.Splitkey=signal

Table A.567: ForbiddenSignalPath

Class	**Frame** (abstract)			
Note	Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*AbstractEthernetFrame*, CanFrame, FlexrayFrame, *LinFrame*			
Aggregated by	ARPackage.element			
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:** atp.Splitkey=pduToFrameMapping, pduToFrame Mapping.shortName, pduToFrameMapping.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.568: Frame

Class	**FrameMapping**			
Note	Maps the source frame to the target frame.			
Base	*ARObject*			
Aggregated by	Gateway.frameMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the frame mapping. **Tags:** atp.Splitkey=introduction
sourceFrame	FrameTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceFrame
targetFrame	FrameTriggering	0..1	ref	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetFrame

Table A.569: FrameMapping

Class	**FramePid**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	AssignFrameIdRange.framePid			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
index	Integer	0..1	attr	This attribute is used to order the frame_PIDs. The values of index shall be unique within one AssignFrameIdRange.
pid	PositiveInteger	0..1	attr	Frame_PID value.

Table A.570: FramePid

Class	FramePort			
Note	Connectors reception or send port on the referenced channel referenced by a FrameTriggering.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.571: FramePort

Class	*FrameTriggering* (abstract)			
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	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanFrameTriggering, EthernetFrameTriggering, FlexrayFrameTriggering, LinFrameTriggering			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frame	Frame	0..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. **Tags:** atp.Splitkey=frame
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. **Tags:** atp.Splitkey=framePort
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:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.572: FrameTriggering

Class	FreeFormat			
Note	Representing freely defined data. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *FreeFormatEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
byteValue (ordered)	Integer	*	attr	The integer Value of a freely defined data byte.

Table A.573: FreeFormat

Class	FunctionInhibitionNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.574: FunctionInhibitionNeeds

Class	Gateway			
Note	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. **Tags:** atp.recommendedPackage=Gateways			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecu	EcuInstance	0..1	ref	Reference to one ECU instance that implements the gateway. **Tags:** atp.Splitkey=ecu
frameMapping	FrameMapping	*	aggr	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. atpVariation: If frames are variable in clusters, the gateway frame mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=frameMapping, frameMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
iPduMapping	IPduMapping	*	aggr	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. atpVariation: If PDUs are variable in clusters, the gateway PDU mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduMapping, iPduMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
signalMapping	ISignalMapping	*	aggr	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. atpVariation: If signals are variable in clusters, the gateway signal mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=signalMapping, signalMapping.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.575: Gateway

Class	GeneralPurposeConnection			
Note	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. **Tags:** atp.recommendedPackage=GeneralPurposeConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pduTriggering	PduTriggering	*	ref	Reference to PduTriggerings that are connected to each other by a GeneralPurposeConnection. **Tags:** atp.Splitkey=pduTriggering

Table A.576: GeneralPurposeConnection

Class	GeneralPurposeIPdu			
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	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.577: GeneralPurposeIPdu

Class	GeneralPurposePdu			
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	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.578: GeneralPurposePdu

Class	GlobalTimeCanMaster			
Note	This represents the specialization of the GlobalTimeMaster for the CAN communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
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.
sync Confirmation Timeout	TimeValue	0..1	attr	This represents the value for the confirmation timeout. Unit: seconds. **Tags:** atp.Status=obsolete

Table A.579: GlobalTimeCanMaster

Class	GlobalTimeCanSlave			
Note	This represents the specialization of the GlobalTimeSlave for the CAN communication.			
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.slave			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcValidated	GlobalTimeCrc ValidationEnum	0..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 messages.

Table A.580: GlobalTimeCanSlave

Class	GlobalTimeCouplingPortProps			
Note	Defines properties for the usage of the CouplingPort in the scope of Global Time Sync.			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.globalTimeProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
propagation Delay	TimeValue	0..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 A.581: GlobalTimeCouplingPortProps

Class	GlobalTimeDomain			
Note	This represents the ability to define a global time domain. **Tags:** atp.recommendedPackage=GlobalTimeDomains			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clonedFrom TimeDomain	GlobalTimeDomain	*	ref	Reference to a set of GlobalTimeDomains from which the time information of the reference source GlobalTime Domain is cloned. This allows to model the cloning source of global time information. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clonedFromTimeDomain, clonedFromTime Domain.globalTimeDomain, clonedFromTime Domain.variationPoint.shortLabel vh.latestBindingTime=postBuild
debounceTime	TimeValue	0..1	attr	Defines the minimum amount of time between two time sync messages are transmitted.
domainId	PositiveInteger	0..1	attr	This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

▽

△

Class	**GlobalTimeDomain**			
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:** atp.Splitkey=gateway, gateway.shortName, gateway.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTime CorrectionProps	GlobalTimeCorrection Props	0..1	aggr	Defintion of attributes for rate and offset correction. **Tags:** atp.Splitkey=globalTimeCorrectionProps
globalTime Domain Property	AbstractGlobalTime DomainProps	0..1	aggr	Additional properties of the GlobalTimeDomain. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomainProperty, globalTime DomainProperty.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTime Master	GlobalTimeMaster	0..1	aggr	This represents the single master of a GlobalTime Domain. A GlobalTimeDomain may have no GlobalTime Domain.master, e.g. when it gets its time from a GPS receiver. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeMaster, globalTimeMaster.short Name, globalTimeMaster.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTimeSub Domain	GlobalTimeDomain	*	ref	By this means it is possible to create a hierarchy of sub Domains where one global time domain can declare one or more other global time domains as its subDomains. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeSubDomain, globalTimeSub Domain.globalTimeDomain, globalTimeSub Domain.variationPoint.shortLabel vh.latestBindingTime=postBuild
icvFreshness ValueId	PositiveInteger	0..1	attr	This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.
icvSecureCom Props	SecOcSecureCom Props	0..1	ref	Reference to a SecureComProps definition to be used for the Integrity Check Value (ICV) calculation and verification. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=icvSecureComProps, icvSecureCom Props.secOcSecureComProps, icvSecureCom Props.variationPoint.shortLabel vh.latestBindingTime=postBuild
maxProgression Mismatch Threshold	TimeValue	0..1	attr	This attribute defines the maximum allowed difference between local time and fallback time of the time base in seconds.
network SegmentId	NetworkSegment Identification	0..1	aggr	Defines the numerical identification of a GlobalTime sub domain. **Tags:** atp.Splitkey=networkSegmentId

▽

△

Class	**GlobalTimeDomain**			
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:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
slave	GlobalTimeSlave	*	aggr	This represents the collections of slaves of the Global TimeDomain. A GlobalTimeDomain may have no Global TimeDomain.slaves, e.g. when it propagates its time directly to sub domains. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=slave, slave.shortName, slave.variation Point.shortLabel vh.latestBindingTime=postBuild
syncLoss Timeout	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 A.582: GlobalTimeDomain

Class	**GlobalTimeEthMaster**			
Note	This represents the specialization of the GlobalTimeMaster for Ethernet communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
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.
holdOverTime	TimeValue	0..1	attr	This attribute defines the timeout for transmission of Sync and Follow_Up messages on Master ports in absence of reception of Sync and Follow_Up messages on Slave port.
subTlvConfig	EthTSynSubTlvConfig	0..1	aggr	Defines the subTLV fields which shall be included in the time sync message. **Tags:** atp.Splitkey=subTlvConfig

Table A.583: GlobalTimeEthMaster

Class	**GlobalTimeFrMaster**			
Note	This represents the specialization of the GlobalTimeMaster for Flexray communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
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.

Table A.584: GlobalTimeFrMaster

Class	**GlobalTimeFrSlave**			
Note	This represents the specialization of the GlobalTimeSlave for Flexray communication.			
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.slave			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
crcValidated	GlobalTimeCrc ValidationEnum	0..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 messages.

Table A.585: GlobalTimeFrSlave

Class	**GlobalTimeGateway**			
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*			
Aggregated by	GlobalTimeDomain.gateway			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
host	EcuInstance	0..1	ref	The GlobalTimeGateway is hosted by the referenced Ecu Instance. **Tags:** atp.Splitkey=host This Attribute is only used by the AUTOSAR Classic Platform.
master	GlobalTimeMaster	0..1	ref	This represents the master of the global time gateway. **Tags:** atp.Splitkey=master
slave	GlobalTimeSlave	0..1	ref	This represents the slave of the GlobalTimeGateway. **Tags:** atp.Splitkey=slave

Table A.586: GlobalTimeGateway

Class	***GlobalTimeMaster*** (abstract)			
Note	This represents the generic concept of a global time master.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	GlobalTimeCanMaster, GlobalTimeEthMaster, GlobalTimeFrMaster, UserDefinedGlobalTimeMaster			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	Communication Connector	0..1	ref	The GlobalTimeMaster is bound to the Communication Connector. **Tags:** atp.Splitkey=communicationConnector
icvSecured	GlobalTimeIcvSupport Enum	0..1	attr	Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages. This Attribute is only used by the AUTOSAR Classic Platform.
immediate ResumeTime	TimeValue	0..1	attr	Defines the minimum time between an "immediate" message and the next periodic message.
isSystemWide GlobalTime Master	Boolean	0..1	attr	If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.
syncPeriod	TimeValue	0..1	attr	This represents the period. Unit: seconds

Table A.587: GlobalTimeMaster

Enumeration	**GlobalTimePortRoleEnum**
Note	Selection of port behavior to Time Slave, Time Master or Dynamic (Time Slave or Time Master at runtime).
Aggregated by	EthGlobalTimeManagedCouplingPort.globalTimePortRole
Literal	***Description***
dynamic	Time Slave or Time Master port behavior at runtime. **Tags:** atp.EnumerationLiteralIndex=2
timeMaster	timeMaster port behavior **Tags:** atp.EnumerationLiteralIndex=1
timeSlave	TimeSlave port behavior **Tags:** atp.EnumerationLiteralIndex=0

Table A.588: GlobalTimePortRoleEnum

Class	***GlobalTimeSlave*** (abstract)			
Note	This represents the generic concept of a global time slave.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	GlobalTimeCanSlave, GlobalTimeEthSlave, GlobalTimeFrSlave, UserDefinedGlobalTimeSlave			
Aggregated by	GlobalTimeDomain.slave			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	Communication Connector	0..1	ref	The GlobalTimeSlave is bound to the Communication Connector. **Tags:** atp.Splitkey=communicationConnector
followUp TimeoutValue	TimeValue	0..1	attr	Rx timeout for the follow-up message.
icvVerification	GlobalTimeIcv VerificationEnum	0..1	attr	Defines how an Integrity Check Value (ICV) shall be handled at the receiver. This Attribute is only used by the AUTOSAR Classic Platform.
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 A.589: GlobalTimeSlave

Enumeration	**HandleInvalidEnum**
Note	Strategies of handling the reception of invalidValue.
Aggregated by	InvalidationPolicy.handleInvalid, ISignalPort.handleInvalid
Literal	***Description***
dontInvalidate	Invalidation is switched off. **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**HandleInvalidEnum**
external Replacement	Replace a received invalidValue. The replacement value is sourced from the aggregation in the role replaceWith. **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 A.590: HandleInvalidEnum

Enumeration	**HandleOutOfRangeEnum**
Note	A value of this type is taken for controlling the range checking behavior of the AUTOSAR RTE.
Aggregated by	ISignalProps.handleOutOfRange, *ReceiverComSpec*.handleOutOfRange, *SenderComSpec*.handleOutOfRange
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 A.591: HandleOutOfRangeEnum

Class	**HardwareConfiguration**			
Note	Describes in which mode the hardware is operating while needing this resource consumption.			
Base	*ARObject*			
Aggregated by	*ExecutionTime*.hardwareConfiguration, *HeapUsage*.hardwareConfiguration, *StackUsage*.hardwareConfiguration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
additional Information	String	0..1	attr	Specifies additional information on the Hardware Configuration.
processorMode	String	0..1	attr	Specifies in which mode the processor is operating.
processor Speed	String	0..1	attr	Specifies the speed the processor is operating.

Table A.592: HardwareConfiguration

Class	**HwAttributeDef**			
Note	This metaclass represents the ability to define a particular hardware attribute. The category of this element defines the type of the attributeValue. If the category is Enumeration the hw AttributeEnumerationLiterals specify the available literals.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	HwCategory.hwAttributeDef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hwAttribute Literal	HwAttributeLiteralDef	*	aggr	The available EnumerationLiterals of the Enumeration definition. Only applicable if the category of the Hw AttributeDef equals Enumeration. **Tags:** atp.Splitkey=hwAttributeLiteral, hwAttribute Literal.shortName
isRequired	Boolean	0..1	attr	This attribute specifies if the defined attribute value is required to be provided.
unit	Unit	0..1	ref	This association specifies the physical unit of the defined hardware attribute. This is optional due to the fact that there are textual attributes. **Tags:** atp.Splitkey=unit

Table A.593: HwAttributeDef

Class	**HwAttributeValue**			
Note	This metaclass represents the ability to assign a hardware attribute value. Note that v and vt are mutually exclusive.			
Base	*ARObject*			
Aggregated by	*HwDescriptionEntity*.hwAttributeValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
annotation	Annotation	0..1	aggr	Optional annotation that can be added to each Hw AttributeValue. **Tags:** atp.Splitkey=annotation
hwAttributeDef	HwAttributeDef	0..1	ref	This association represents the definition of the particular hardware attribute value. **Tags:** atp.Splitkey=hwAttributeDef
v	Numerical	0..1	attr	This represents a numerical hardware attribute value. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
vt	VerbatimString	0..1	attr	This represents a textual hardware attribute value.

Table A.594: HwAttributeValue

Class	***HwDescriptionEntity*** (abstract)			
Note	This meta-class represents the ability to describe a hardware entity.			
Base	*ARObject*, *Referrable*			
Subclasses	HwElement, HwPin, HwPinGroup, HwType			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***HwDescriptionEntity*** (abstract)			
hwAttribute Value	HwAttributeValue	*	aggr	This aggregation represents a particular hardware attribute value. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwAttributeValue, hwAttributeValue.variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=50
hwCategory	HwCategory	*	ref	One of the associations representing one particular category of the hardware entity. **Tags:** atp.Splitkey=hwCategory xml.sequenceOffset=30
hwType	HwType	0..1	ref	This association is used to assign an optional HwType which contains the common attribute values for all occurences of this HwDescriptionEntity. Note that Hw Types can not be redefined and therefore shall not have a hwType reference. **Tags:** atp.Splitkey=hwType

Table A.595: HwDescriptionEntity

Class	**HwElement**			
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	*ARElement*, *ARObject*, *CollectableElement*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hwElement Connection	HwElementConnector	*	aggr	This represents one particular connection between two hardware elements. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwElementConnection, hwElement Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=110
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:** atp.Splitkey=hwPinGroup, hwPinGroup.shortName, hw PinGroup.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=90

▽

△

Class	**HwElement**			
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:** atp.Splitkey=nestedElement, nestedElement.hwElement, nestedElement.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=70

Table A.596: HwElement

Class	**HwElementConnector**			
Note	This meta-class represents the ability to connect two hardware elements. The details of the connection can be refined by hwPinGroupConnection.			
Base	*ARObject*, *Describable*			
Aggregated by	HwElement.hwElementConnection			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
hwElement	HwElement	*	ref	This association connects two hardware elements. **Tags:** atp.Splitkey=hwElement
hwPin Connection	HwPinConnector	*	aggr	This represents one particular connection between two hardware pins. This connection shall be used if pin-to-pin-connection is to be described but no description of the connection between the hierarchical composition of HwPinGroups (using HwPinGroupConnector) is required. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwPinConnection, hwPin Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=60
hwPinGroup Connection	HwPinGroupConnector	*	aggr	This represents one particular connection between two hardware pin groups. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwPinGroupConnection, hwPinGroup Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=50

Table A.597: HwElementConnector

Class	**HwPinConnector**			
Note	This meta-class represents the ability to connect two pins.			
Base	*ARObject*, *Describable*			
Aggregated by	HwElementConnector.hwPinConnection, HwPinGroupConnector.hwPinConnection			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
hwPin	HwPin	*	ref	This association connects two hardware pins. **Tags:** atp.Splitkey=hwPin

Table A.598: HwPinConnector

Class	HwPinGroup			
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	*ARObject*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	HwElement.hwPinGroup, HwPinGroupContent.hwPinGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hwPinGroup Content	HwPinGroupContent	0..1	aggr	This aggregation describes the contained pins/pin groups. **Tags:** atp.Splitkey=hwPinGroupContent

Table A.599: HwPinGroup

Class	HwPinGroupConnector			
Note	This meta-class represents the ability to connect two pin groups.			
Base	*ARObject*, *Describable*			
Aggregated by	HwElementConnector.hwPinGroupConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hwPin Connection	HwPinConnector	*	aggr	This represents one particular connection between two hardware pins. The connected pins shall match the connection provided by the parent hwPinGroup Connection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwPinConnection, hwPin Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
hwPinGroup	HwPinGroup	*	ref	This association connects two hardware pin groups. **Tags:** atp.Splitkey=hwPinGroup

Table A.600: HwPinGroupConnector

Class	HwPortMapping			
Note	HWPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.			
Base	*ARObject*			
Aggregated by	ECUMapping.hwPortMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Connector	Communication Connector	0..1	ref	Reference to the CommunicationConnector in the System Template **Tags:** atp.Splitkey=communicationConnector
hw Communication Port	HwPinGroup	0..1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the Hw CommunicationController is described in the Ecu Resource Description. **Tags:** atp.Splitkey=hwCommunicationPort

Table A.601: HwPortMapping

Class	HwType			
Note	This represents the ability to describe Hardware types on an abstract 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=HwTypes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.602: HwType

Class	***IEEE1722TpAcfBus*** (abstract)			
Note	Abstract class to define various busses to be transported over a IEEE1722TP ACF connection. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	IEEE1722TpAcfCan, IEEE1722TpAcfLin			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
acfPart	IEEE1722TpAcfBusPart	*	aggr	One part transported over IEEE1722Tp channel. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=acfPart, acfPart.shortName, acfPart.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
busId	PositiveInteger	0..1	attr	Id of the transported bus over the ACF connection.

Table A.603: IEEE1722TpAcfBus

Class	**IEEE1722TpAcfCan**			
Note	ACF IEEE1722Tp bus used for CAN transport. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARObject*, *IEEE1722TpAcfBus*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
messageType	IEEE1722TpAcfCanMessageTypeEnum	0..1	attr	Definition of the ACF CAN stream message type.

Table A.604: IEEE1722TpAcfCan

Class	**IEEE1722TpAcfCanPart**			
Note	Definition of one CAN part (frame or frame range) transported over the IEEE1722Tp channel. **Tags:** atp.Status=candidate			
Base	*ARObject*, *IEEE1722TpAcfBusPart*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*IEEE1722TpAcfBus*.acfPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**IEEE1722TpAcfCanPart**			
canAddressing Mode	CanAddressingMode Type	0..1	attr	Defines whether standard or extended address format shall be used.
canBitRate Switch	Boolean	0..1	attr	Defines whether the bit rate switch bit shall be set.
canFrameTx Behavior	CanFrameTxBehavior Enum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
canIdentifier	PositiveInteger	0..1	attr	Optional Can Id defined in case the Can Id can not be determined during runtime.
canIdentifier Mask	PositiveInteger	0..1	attr	CAN identifier mask which denotes relevant bits in the CAN Identifier. This attribute defines a CAN Identifier range in an alternative way to canIdentifierRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier.
canIdentifier Range	RxIdentifierRange	0..1	aggr	Definition of the identifier range for IEEE1722Tp ACF Can messages. **Tags:** atp.Splitkey=canIdentifierRange atp.Status=candidate
sdu	PduTriggering	0..1	ref	Reference to the Pdu transported in the IEEE1722Tp channel. **Tags:** atp.Splitkey=sdu atp.Status=candidate

Table A.605: IEEE1722TpAcfCanPart

Class	**IEEE1722TpAcfConnection**			
Note	ACF IEEE1722Tp connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
acfMaxTransit Time	TimeValue	0..1	attr	Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.
acfTransported Bus	IEEE1722TpAcfBus	*	aggr	Definition of the transported busses over this ACF connection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=acfTransportedBus, acfTransportedBus.shortName, acfTransportedBus.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
collection Threshold	PositiveInteger	0..1	attr	Defines the size threshold in bytes which, when exceeded, triggers the sending of the IEEE1722Tp ACF message, even when the maxium IEEE1722Tp ACF message size has not been reached yet.

▽

△

Class	**IEEE1722TpAcfConnection**			
collection Timeout	TimeValue	0..1	attr	When this timeout expires the IEEE1722Tp ACF message is triggered for sending. The respective timer is started when the first Pdu is put into the IEEE1722Tp ACF message. Defined in seconds.
mixedBusType Collection	Boolean	0..1	attr	Defines if this ACF-stream is allowed to collect ACF-messages of different bus kinds (i.e. whether it is allowed to collect CAN and LIN ACF-messages in one ACF-stream message).

Table A.606: IEEE1722TpAcfConnection

Class	**IEEE1722TpAcfLin**			
Note	ACF IEEE1722Tp bus used for LIN transport. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARObject*, *IEEE1722TpAcfBus*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseFrequency	PositiveInteger	0..1	attr	CRF base frequency in Hz.
frameSync Enabled	Boolean	0..1	attr	Defines whether the "fs" (frame sync) shall be enabled.
timestamp Interval	PositiveInteger	0..1	attr	CRF timestamp interval as multiple of the baseFrequency.

Table A.607: IEEE1722TpAcfLin

Class	**IEEE1722TpAcfLinPart**			
Note	Definition of one LIN part transported over the IEEE1722Tp channel. **Tags:** atp.Status=candidate			
Base	*ARObject*, *IEEE1722TpAcfBusPart*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*IEEE1722TpAcfBus*.acfPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
linIdentifier	PositiveInteger	0..1	attr	Optional Lin Id defined in case the Lin Id can not be determined during runtime. **Tags:** atp.Status=candidate
sdu	PduTriggering	0..1	ref	Reference to the Pdu transported in the IEEE1722Tp channel. **Tags:** atp.Splitkey=sdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table A.608: IEEE1722TpAcfLinPart

Class	***IEEE1722TpAvConnection*** (abstract)			
Note	AV IEEE1722Tp connection. **Tags:** atp.Status=candidate			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpConnection*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	IEEE1722TpAafConnection, IEEE1722TpCrfConnection, IEEE1722TplidcConnection, IEEE1722TpRvf Connection			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxTransitTime	TimeValue	0..1	attr	Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.
sdu	PduTriggering	*	ref	Reference to the upper layer Sdu used for the transport of the payload of the IEEE1722Tp. **Tags:** atp.Splitkey=sdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table A.609: IEEE1722TpAvConnection

Class	***IEEE1722TpConnection*** (abstract)			
Note	Definition of the IEEE1722Tp protocol. **Tags:** atp.Status=candidate			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	IEEE1722TpAcfConnection, *IEEE1722TpAvConnection*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationMac Address	MacAddressString	0..1	attr	Optional definition of the destination MAC address for this stream. If no given then macAddressStreamId is used as destination MAC address. **Tags:** atp.Status=candidate
globalTime Domain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this IEEE1722Tp Connection shall be synchronized with. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTime Domain.globalTimeDomain, globalTimeDomain.variation Point.shortLabel vh.latestBindingTime=systemDesignTime
macAddress StreamId	MacAddressString	0..1	attr	MAC Address part of the Stream Id. **Tags:** atp.Status=candidate
pdu	PduTriggering	0..1	ref	Reference to the lower layer Pdu used for the IEEE1722 Tp protocol transport. **Tags:** atp.Splitkey=pdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	***IEEE1722TpConnection*** (abstract)			
uniqueStreamId	PositiveInteger	0..1	attr	Unique Id part of the Stream Id. **Tags:** atp.Status=candidate
version	PositiveInteger	0..1	attr	Version of the IEEE1722TP stream. **Tags:** atp.Status=candidate
vlanPriority	PositiveInteger	0..1	attr	Optional definition of the VLAN priority for this stream.

Table A.610: IEEE1722TpConnection

Class	**IEEE1722TplidcConnection**			
Note	AV IEEE1722Tp IIDC and IEC61883 connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpAvConnection*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iidcChannel	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 channel. **Tags:** atp.Status=candidate
iidcDataBlock Size	PositiveInteger	0..1	attr	Definition of the IEC61883 data block size (DBS). **Tags:** atp.Status=candidate
iidcFraction Number	PositiveInteger	0..1	attr	Definition of the IEC61883 fractionNumber (FN). **Tags:** atp.Status=candidate
iidcSource PacketHeader	Boolean	0..1	attr	Defines the IEC61883 source packet header (SPH) existence. **Tags:** atp.Status=candidate
iidcStream Format	PositiveInteger	0..1	attr	Definition of the IEC61883 stream format (FMT). **Tags:** atp.Status=candidate
iidcSy	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 sy. **Tags:** atp.Status=candidate
iidcTag	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 tag. **Tags:** atp.Status=candidate
iidcTCode	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 tcode. **Tags:** atp.Status=candidate

Table A.611: IEEE1722TplidcConnection

Class	**IPSecRule**			
Note	This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPSecConfig.ipSecRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
direction	Communication DirectionType	0..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.

▽

△

Class	**IPSecRule**			
headerType	IPsecHeaderTypeEnum	0..1	attr	Header type specifying the IPsec security mechanism.
ipProtocol	IPsecIpProtocolEnum	0..1	attr	This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.
localCertificate	CryptoService Certificate	*	ref	This reference identifies the applicable certificate used for a local authentication. **Tags:** atp.Splitkey=localCertificate
localId	String	0..1	attr	This attribute defines how the local participant should be identified for authentication.
localIkePq PresharedKey Support	LocalPqPresharedKey Support	0..1	aggr	Post-quantum pre-shared key configuration to support IETF RFC 8784 to ensure resistance against post-quantum attacks by combining a post-quantum pre-shared key (PPK) with the exchanged information (via e.g., DH/ECDH) during key derivation. **Tags:** atp.Splitkey=localIkePqPresharedKeySupport
localPortRange End	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an 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 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	0..1	attr	This attribute defines the type of the connection.
policy	IPsecPolicyEnum	0..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.
preSharedKey	CryptoServiceKey	0..1	ref	This reference identifies the applicable cryptographic key used for authentication. **Tags:** atp.Splitkey=preSharedKey
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".
remote Certificate	CryptoService Certificate	*	ref	This reference identifies the applicable certificate used for a remote authentication. **Tags:** atp.Splitkey=remoteCertificate
remoteId	String	0..1	attr	This attribute defines how the remote participant should be identified for authentication.
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. **Tags:** atp.Splitkey=remoteIpAddress

▽

△

Class	IPSecRule			
remotePort RangeEnd	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an 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 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 A.612: IPSecRule

Class	*IPdu* (abstract)			
Note	The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	ContainerIPdu, DcmIPdu, GeneralPurposeIPdu, ISignalIPdu, J1939DcmIPdu, J1939ProtectedIPdu, MultiplexedIPdu, NPdu, SecuredIPdu, UserDefinedIPdu			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.613: IPdu

Class	IPduMapping			
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	*ARObject*			
Aggregated by	Gateway.iPduMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the IPdu mapping. **Tags:** atp.Splitkey=introduction
pduMaxLength	PositiveInteger	0..1	attr	Define the maximum length in bytes which limits the length of the Pdu during gateway operation if the runtime length of the received Pdu exceeds this limit.
pdurTpChunk Size	PositiveInteger	0..1	attr	Optionally defines the to be configured Pdu Router Tp ChunkSize for this routing relation.
sourceIPdu	PduTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceIPdu
targetIPdu	TargetIPduRef	0..1	aggr	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetIPdu

Table A.614: IPduMapping

Class	IPduPort			
Note	Connectors reception or send port on the referenced channel referenced by a PduTriggering.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
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 A.615: IPduPort

Class	IPv6ExtHeaderFilterList			
Note	Permitted list for the filtering of IPv6 extension headers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPv6ExtHeaderFilterSet.extHeaderFilterList			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allowedIPv6Ext Header	PositiveInteger	*	attr	IPv6 Extension Header type allowed by this filter.

Table A.616: IPv6ExtHeaderFilterList

Class	ISignal			
Note	Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup. **Tags:** atp.recommendedPackage=ISignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**ISignal**			
data Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataTransformation, data Transformation.dataTransformation, data Transformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	0..1	attr	With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "Init Value". If a full DataMapping exist for the SystemSignal this information may be available from a configured Sender ComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification. **Tags:** atp.Splitkey=initValue
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Tags:** atp.Splitkey=iSignalProps
iSignalType	ISignalTypeEnum	0..1	attr	This attribute defines whether this iSignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type.
length	UnlimitedInteger	0..1	attr	Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.

▽

△

Class	**ISignal**			
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalid Value" and the Data Semantics. **Tags:** atp.Splitkey=networkRepresentationProps
systemSignal	SystemSignal	0..1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal. **Tags:** atp.Splitkey=systemSignal
timeout Substitution Value	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstituition for this ISignal. **Tags:** atp.Splitkey=timeoutSubstitutionValue
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table A.617: ISignal

Class	**ISignalGroup**			
Note	SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage) **Tags:** atp.recommendedPackage=ISignalGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ISignalGroup**			
comBased SignalGroup Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation, comBasedSignalGroupTransformation.dataTransformation, comBasedSignalGroupTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together. **Tags:** atp.Splitkey=iSignal
systemSignal Group	SystemSignalGroup	0..1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup. **Tags:** atp.Splitkey=systemSignalGroup
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignal Groups are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table A.618: ISignalGroup

Class	**ISignalIPdu**			
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	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduTiming Specification	IPduTiming	0..1	aggr	Timing specification for Com IPdus (Transmission Modes). This information is mandatory for the sender in a System Extract. This information may be omitted on receivers in a System Extract. atpVariation: The timing of a Pdu can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduTimingSpecification, iPduTimingSpecification.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**ISignalIPdu**			
iSignalToPdu Mapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the Signal IPdu. atpVariation: The content of a PDU can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalToPduMapping, iSignalToPdu Mapping.shortName, iSignalToPduMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.

Table A.619: ISignalIPdu

Class	**ISignalIPduGroup**			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups. **Tags:** atp.recommendedPackage=ISignalIPduGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..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	0..1	attr	This attribute defines the use-case for this ISignalIPdu 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 ISignalIPdu Group	ISignalIPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance. **Tags:** atp.Splitkey=containedISignalIPduGroup
iSignalIPdu	ISignalIPdu	*	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:** atp.Splitkey=iSignalIPdu, iSignalIPdu.iSignalIPdu, iSignal IPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmPdu, nmPdu.nmPdu, nmPdu.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.620: ISignalIPduGroup

Class	**ISignalMapping**			
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*			
Aggregated by	Gateway.signalMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ISignal mapping. **Tags:** atp.Splitkey=introduction
sourceSignal	ISignalTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceSignal
targetSignal	ISignalTriggering	0..1	ref	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetSignal

Table A.621: ISignalMapping

Class	**ISignalPort**			
Note	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.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	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 DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec. In this case the ReceiverComSpec overrides this optional specification. **Tags:** atp.Splitkey=dataFilter
ddsQosProfile	DdsCpQosProfile	0..1	ref	Reference to the DDS Qos profile used for this ISignal. **Tags:** atp.Splitkey=ddsQosProfile atp.Status=candidate
firstTimeout	TimeValue	0..1	attr	• 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.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.

△

Class	**ISignalPort**			
timeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverComSpec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. • ISignalPort with communicationDirection = out: Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the SenderComSpec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the System Signal and the value is available in the configured Sender ComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. This attribute can be used in the following cases: • legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the DataMapping is missing. • bus monitoring use cases in which the DataMapping is ignored.

Table A.622: ISignalPort

Class	**ISignalPortToDiagnosticEventMapping**			
Note	This mapping is used to identify the DiagnosticEvents that are reported if during the reception of an ISignalGroup on a specific Ecu an error is detected. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comTimeout Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a com timeout error (ISignalGroup is not received within expected time) occurred. **Tags:** atp.Splitkey=comTimeoutError xml.sequenceOffset=10
dataValueError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a data value error (invalid or out of range) occurred. **Tags:** atp.Splitkey=dataValueError xml.sequenceOffset=60

▽

△

Class	**ISignalPortToDiagnosticEventMapping**			
e2eCounter Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E CRC counter error occurred. **Tags:** atp.Splitkey=e2eCounterError xml.sequenceOffset=30
e2eCrcError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E CRC error occurred. **Tags:** atp.Splitkey=e2eCrcError xml.sequenceOffset=20
e2eTimeout Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E timeout error occurred. **Tags:** atp.Splitkey=e2eTimeoutError xml.sequenceOffset=40
iSignalPort	ISignalPort	0..1	ref	Reference to an ISignalPort on which the ISignalGroup is received. **Tags:** atp.Splitkey=iSignalPort xml.sequenceOffset=110
iSignal Triggering	ISignalTriggering	0..1	ref	Reference to an ISignalTriggering of an ISignalGroup that is received on the ISignalPort. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.iSignal Triggering, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=100
secureOnboard ComError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a SecOc error occurred. **Tags:** atp.Splitkey=secureOnboardComError xml.sequenceOffset=50

Table A.623: ISignalPortToDiagnosticEventMapping

Class	**ISignalToIPduMapping**			
Note	An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ISignalIPdu.iSignalToPduMapping, NmPdu.iSignalToIPduMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iSignal	ISignal	0..1	ref	Reference to a ISignal that is mapped into the ISignal IPdu. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive. **Tags:** atp.Splitkey=iSignal

▽

△

Class	**ISignalToIPduMapping**			
iSignalGroup	ISignalGroup	0..1	ref	Reference to an ISignalGroup that is mapped into the SignalIPdu. If an ISignalToIPduMapping for an ISignal Group is defined, only the UpdateIndicationBitPosition and the transferProperty is relevant. The startPosition and the packingByteOrder shall be ignored. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive. **Tags:** atp.Splitkey=iSignalGroup
packingByte Order	ByteOrderEnum	0..1	attr	This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.
startPosition	UnlimitedInteger	0..1	attr	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.
transferProperty	TransferPropertyEnum	0..1	attr	Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.
update IndicationBit Position	UnlimitedInteger	0..1	attr	The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table A.624: ISignalToIPduMapping

Class	**ISignalTriggering**			
Note	A ISignalTriggering allows an assignment of ISignals to physical channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.iSignalTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iSignal	ISignal	0..1	ref	This reference shall be used if an ISignal is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignalTriggering-ISignalGroup reference. **Tags:** atp.Splitkey=iSignal
iSignalGroup	ISignalGroup	0..1	ref	This reference shall be used if an ISignalGroup is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignal Triggering-ISignal reference. **Tags:** atp.Splitkey=iSignalGroup
iSignalPort	ISignalPort	*	ref	References to the ISignalPort on every ECU of the system which sends and/or receives the ISignal. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=iSignalPort

Table A.625: ISignalTriggering

Enumeration	**ISignalTypeEnum**
Note	This enumeration defines ISignal types that are used for derivation of the ComSignalType in the COM configuration.
Aggregated by	ISignal.iSignalType
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 A.626: ISignalTypeEnum

Class	***Identifiable*** (abstract)
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	*ARObject*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	***Identifiable*** (abstract)			
Subclasses	ARPackage, *AbstractDoIpLogicAddressProps*, *AbstractEvent*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstanceFilter*, *AbstractServiceInstance*, AppOsTask ProxyToEcuTaskProxyMapping, ApplicationEndpoint, ApplicationError, ApplicationPartitionToEcuPartition Mapping, AppliedStandard, AsynchronousServerCallResultPoint, *AtpBlueprint*, *AtpClassifier*, *Atp Feature*, AutosarOperationArgumentInstance, AutosarVariableInstance, *BinaryManifestAddressable Object*, BinaryManifestItemDefinition, *BinaryManifestResource*, BinaryManifestResourceDefinition, Block State, BswInternalTriggeringPoint, BswModuleDependency, *BuildActionEntity*, BuildActionEnvironment, CanSecSecureChannel, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClientIdDefinition, Client ServerOperation, Code, *CollectableElement*, ComManagementMapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEvent Group, *CouplingElementAbstractDetails*, CouplingPort, *CouplingPortAbstractShaper*, *CouplingPort StructuralElement*, *CpSoftwareClusterResource*, CpSoftwareClusterResourceToApplicationPartition Mapping, CpSoftwareClusterToApplicationPartitionMapping, CpSoftwareClusterToEcuInstanceMapping, CpSoftwareClusterToResourceMapping, *CryptoServiceMapping*, CyclicHandlingComDataToOsTask ProxyMapping, DataPrototypeGroup, DataPrototypeTransformationPropsIdent, DataTransformation, *Dds AbstractServiceInstanceElementCp*, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, Diagnostic ExtendedDataRecordElement, DiagnosticFunctionInhibitSource, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, DltArgumentProps, DltLogChannel, Dlt Message, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, ECUMapping, *EOCExecutableEntity RefAbstract*, EcuPartition, EcuPartitionToCoreMapping, EcucContainerValue, *EcucDefinitionElement*, EcucDestinationUriDef, EcucEnumerationLiteralDef, EcucQuery, EcucValidationCondition, Ethernet WakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeature Relation, FMFeatureRestriction, FMFeatureSelection, FlatInstanceDescriptor, FlexrayTpConnection Control, FlexrayTpNode, FlexrayTpPduPool, *FrameTriggering*, GlobalTimeGateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, *IEEE1722 TpAcfBus*, *IEEE1722TpAcfBusPart*, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignal Triggering, *IdentCaption*, ImpositionTime, InternalTriggeringPoint, J1939Node, J1939SharedAddress Cluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacAddress VlanMembership, MacMulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, Mode Declaration, ModeDeclarationMapping, ModeSwitchPoint, ModeSwitchSenderComSpecProps, Network Endpoint, *NmCluster*, NmEcu, *NmNode*, NvBlockDescriptor, *PackageableElement*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, *Physical Channel*, PortElementToCommunicationResourceMapping, PortGroup, *PortInterfaceMapping*, Queued ReceiverComSpecProps, ResourceConsumption, RootSwCompositionPrototype, RptComponent, Rpt Container, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptService Point, RteEventInCompositionSeparation, RteEventInCompositionToOsTaskProxyMapping, RteEventIn SystemSeparation, RteEventInSystemToOsTaskProxyMapping, RunnableEntityGroup, *SdgAttribute*, Sdg Class, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunication FreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, *ServerCallPoint*, ServerComSpecProps, *ServiceNeeds*, SignalServiceTranslationElementProps, SignalServiceTranslation EventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, *StackUsage*, Static SocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplicationPartitionMapping, SwcToEcuMapping, SwcToImplMapping, SwitchAsynchronousTraffic ShaperGroupEntry, SwitchAtsInstanceEntry, SwitchFlowMeteringEntry, SwitchStreamFilterActionDest PortModification, SwitchStreamFilterEntry, SwitchStreamFilterRule, SwitchStreamGateEntry, Switch StreamIdentification, SystemMapping, SystemSignalGroupToCommunicationResourceMapping, System SignalToCommunicationResourceMapping, TDCpSoftwareClusterMapping, TDCpSoftwareCluster ResourceMapping, TcpOptionFilterList, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, TimingModeInstance, TlsCryptoCipher Suite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, TraceableTable, TraceableText, *TracedFailure*, TransformationISignalPropsIdent, *TransformationProps*, TransformationTechnology, Trigger, Variable Access, VariationPointProxy, ViewMap, VlanConfig, WaitPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-40

▽

△

Class	***Identifiable*** (abstract)			
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:** atp.Splitkey=annotation 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	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:** atp.Splitkey=desc 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:** atp.Splitkey=introduction 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 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. **Tags:** xml.attribute=true

Table A.627: Identifiable

Primitive	**Identifier**			
Note	An Identifier is a string with a number of constraints on its appearance, satisfying the requirements typical programming languages define for their Identifiers. This datatype represents a string, that can be used as a c-Identifier. It shall start with a letter, may consist of letters, digits and underscores. **Tags:** xml.xsd.customType=IDENTIFIER xml.xsd.maxLength=128 xml.xsd.pattern=[a-zA-Z][a-zA-Z0-9_]* xml.xsd.type=string			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=draft xml.attribute=true
namePattern	String	0..1	attr	This attribute represents a pattern which shall be used to define the value of the identifier if the identifier in question is part of a blueprint. For more details refer to TPS_StandardizationTemplate. **Tags:** xml.attribute=true

Table A.628: Identifier

Class	**IkePqPresharedKey**			
Note	Post-quantum pre-shared key (PPK) identified by post-quantum pre-shared key identifier (PKK_ID).			
Base	*ARObject*			
Aggregated by	LocalPqPresharedKeySupport.ikePqPresharedKey			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pqPreshared Key	CryptoServiceKey	0..1	ref	Reference to PPK post-quantum pre-shared key. **Tags:** atp.Splitkey=pqPresharedKey
pqPreshared KeyId	PositiveInteger	0..1	attr	This attribute defines the post-quantum pre-shared key identifier for the referenced pqPresharedKey.

Table A.629: IkePqPresharedKey

Class	***Implementation*** (abstract)			
Note	Description of an implementation a single software component or module. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	BswImplementation, SwcImplementation			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***Implementation*** (abstract)			
buildAction Manifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=buildActionManifest, buildAction Manifest.buildActionManifest, buildAction Manifest.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	*	aggr	Specifies the provided implementation code. **Tags:** atp.Splitkey=codeDescriptor, codeDescriptor.short Name
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released **Tags:** atp.Splitkey=compiler, compiler.shortName
generated Artifact	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:** atp.Splitkey=generatedArtifact, generatedArtifact.short Name, generatedArtifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
hwElement	HwElement	*	ref	The hardware elements (e.g. the processor) required for this implementation. **Tags:** atp.Splitkey=hwElement
linker	Linker	*	aggr	Specifies the linker for which this implementation has been released. **Tags:** atp.Splitkey=linker, linker.shortName
mcSupport	McSupportData	0..1	aggr	The measurement & calibration support data belonging to this implementation. The measurement & calibration support data belonging to this implementation. **Tags:** atp.Splitkey=mcSupport
programming Language	Programminglanguage Enum	0..1	attr	Programming language the implementation was created in.
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:** atp.Splitkey=requiredArtifact, requiredArtifact.shortName, requiredArtifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	*Implementation* (abstract)			
required GeneratorTool	DependencyOnArtifact	*	aggr	Relates this Implementation to a generator tool in order to generate additional artifacts during integration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredGeneratorTool, requiredGenerator Tool.shortName, requiredGeneratorTool.variation Point.shortLabel vh.latestBindingTime=preCompileTime
resource Consumption	ResourceConsumption	0..1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class. **Tags:** atp.Splitkey=resourceConsumption, resource Consumption.shortName
swcBsw Mapping	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 Implementtion or for both. **Tags:** atp.Splitkey=swcBswMapping
swVersion	RevisionLabelString	0..1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.
usedCode Generator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	0..1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

Table A.630: Implementation

Class	**ImplementationDataType**			
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*, *AtpClassifier*, *AtpType*, *Autosar DataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
isStructWith Optional Element	Boolean	0..1	attr	This attribute is only valid if the attribute category is set to STRUCTURE. If set to true, this attribute indicates that the ImplementationDataType has been created with the intention to define at least one element of the structure as optional.

▽

△

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 Implementation DataType. **Tags:** atp.Splitkey=symbolProps, symbolProps.shortName
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

Table A.631: ImplementationDataType

Class	**ImplementationDataTypeElement**			
Note	Declares a data object which is locally aggregated. Such an element can only be used within the scope where it is aggregated. This element either consists of further `subElement`s or it is further defined via its `swDataDefProps`. There are several use cases within the system of `ImplementationDataType`s for such a local declaration: • 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	*ARObject*, *AbstractImplementationDataTypeElement*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ImplementationDataType.subElement, ImplementationDataTypeElement.subElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arrayImplPolicy	ArrayImplPolicyEnum	0..1	attr	This attribute controls the implementation of the payload of an array. It shall only be used if the enclosing `ImplementationDataType` constitutes an array.
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.

▽

△

Class	**ImplementationDataTypeElement**			
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.
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs"). 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
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement. **Tags:** atp.Splitkey=swDataDefProps

Table A.632: ImplementationDataTypeElement

Class	**ImplementationDataTypeElementInPortInterfaceRef**			
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	*ARObject*, *DataPrototypeReference*			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElementProps.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context ImplementationDataElement (ordered)	AbstractImplementationDataTypeElement	*	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:** atp.Splitkey=rootDataPrototype xml.sequenceOffset=10
target ImplementationDataType Element	AbstractImplementationDataTypeElement	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:** atp.Splitkey=targetImplementationDataTypeElement xml.sequenceOffset=30

Table A.633: ImplementationDataTypeElementInPortInterfaceRef

Class	**ImplementationDataTypeSubElementRef**			
Note	This meta-class represents the specialization of `SubElementMapping` with respect to [ARMeta Class{ImplementationDataType}}s.			
Base	*ARObject*, *SubElementRef*			
Aggregated by	SubElementMapping.firstElement, SubElementMapping.secondElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
implementation DataType Element	ArVariableIn ImplementationData InstanceRef	0..1	aggr	This represents the referenced implementationDataType Element. **Tags:** atp.Splitkey=implementationDataTypeElement, implementationDataTypeElement.portPrototype, implementationDataTypeElement.rootVariableData Prototype, implementationDataTypeElement.contextData Prototype, implementationDataTypeElement.targetData Prototype
parameter Implementation DataType Element	ArParameterIn ImplementationData InstanceRef	0..1	aggr	This represents the referenced ImplementationDataType Element. **Tags:** atp.Splitkey=parameterImplementationDataType Element, parameterImplementationDataType Element.contextDataPrototype, parameterImplementation DataTypeElement.portPrototype, parameter ImplementationDataTypeElement.rootParameterData Prototype, parameterImplementationDataType Element.targetDataPrototype

Table A.634: ImplementationDataTypeSubElementRef

Class	**ImplementationElementInParameterInstanceRef**			
Note	Describes a reference to a particular ImplementationDataTypeElement instance in the context of a given ParameterDataPrototype. Thus it refers to a particular element in the implementation description of a software data structure. Use Case: The RTE generator publishes its generated structure of calibration parameters in its BSW module description using the "constantMemory" role of ParameterDataPrototypes. Each ParameterData Prototype describes a group of single calibration parameters. In order to point to these single parameters, this "instance ref" is needed. Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataType Element isn't derived from AtpPrototype.			
Base	*ARObject*			
Aggregated by	McDataInstance.instanceInMemory			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context	ParameterData Prototype	0..1	ref	The context for the referred element. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=20
target	AbstractImplementation DataTypeElement	0..1	ref	The referred data element. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=30

Table A.635: ImplementationElementInParameterInstanceRef

Class	***ImplementationProps*** (abstract)
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*

▽

△

Class	***ImplementationProps*** (abstract)			
Subclasses	BswSchedulerNamePrefix, ExecutableEntityActivationReason, SectionNamePrefix, SymbolProps, SymbolicNameProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
symbol	CIdentifier	0..1	attr	The symbol to be used as (depending on the concrete case) either a complete replacement or a prefix.

Table A.636: ImplementationProps

Class	**IndexedArrayElement**			
Note	This element represents exactly one indexed element in the array. Either the applicationArrayElement or implementationArrayElement reference shall be used.			
Base	*ARObject*			
Aggregated by	SenderRecArrayElementMapping.indexedArrayElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationArrayElement	ApplicationArrayElement	0..1	ref	Reference to an ApplicationArrayElement in an array. **Tags:** atp.Splitkey=applicationArrayElement
implementationArrayElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array. **Tags:** atp.Splitkey=implementationArrayElement
index	Integer	0..1	attr	Position of an element in an array. Starting position is 0.

Table A.637: IndexedArrayElement

Class	**InitEvent**			
Note	This `RTEEvent` is supposed to be used for initialization purposes, i.e. for starting and restarting a partition. It is not guaranteed that all `RunnableEntity`s referenced by this `InitEvent` are executed before the 'regular' `RunnableEntity`s are executed for the first time. The execution order depends on the task mapping.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.638: InitEvent

Class	**InitialSdDelayConfig**			
Note	This element is used to configure the offer behavior of the server and the find behavior on the client.			
Base	*ARObject*			
Aggregated by	SomeipSdClientServiceInstanceConfig.initialFindBehavior, SomeipSdServerServiceInstanceConfig.initialOfferBehavior			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialDelayMaxValue	TimeValue	0..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	0..1	attr	Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig).

▽

△

Class	**InitialSdDelayConfig**			
initial RepetitionsBaseDelay	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.
initial RepetitionsMax	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 A.639: InitialSdDelayConfig

Class	**InstantiationDataDefProps**			
Note	This is a general class allowing to apply additional `SwDataDefProps` to particular instantiations of a `DataPrototype`. Typically the accessibility and further information like alias names for a particular data is modeled on the level of `DataPrototype`s (especially `VariableDataPrototype`s, `ParameterDataPrototype`s). But due to the recursive structure of the meta-model concerning data types (a composite (data) type consists out of data prototypes) a part of the MCD information is described in the data type (in case of `ApplicationCompositeDataType`). This is a strong restriction in the reuse of data typed because the data type should be re-used for different `VariableDataPrototype`s and `ParameterDataPrototype`s to guarantee type compatibility on C-implementation level (e.g. data of a Port is stored in PIM or a `ParameterDataPrototype` used as ROM Block and shall be typed by the same data type as NVRAM Block). This class overcomes such a restriction if applied properly.			
Base	*ARObject*			
Aggregated by	NvBlockDescriptor.instantiationDataDefProps, ParameterSwComponentType.instantiationDataDefProps, SwcInternalBehavior.instantiationDataDefProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
parameterInstance	AutosarParameterRef	0..1	aggr	This reference identifies the particular `DataPrototype` (defined in the context of a composite `ParameterDataPrototype`) on which the swDataDefProps shall be applied. **Tags:** atp.Splitkey=parameterInstance
swDataDefProps	SwDataDefProps	0..1	aggr	These are the particular data definition properties which shall be applied **Tags:** atp.Splitkey=swDataDefProps
variableInstance	AutosarVariableRef	0..1	aggr	This reference identifies the particular `DataPrototype` (defined in the context of a composite `VariableDataPrototype`) on which the swDataDefProps shall be applied. **Tags:** atp.Splitkey=variableInstance

Table A.640: InstantiationDataDefProps

Class	***InstantiationRTEEventProps*** (abstract)			
Note	This meta-class represents the ability to refine the properties of `RTEEvent`s for particular instances of a software component.			
Base	*ARObject*			
Subclasses	InstantiationTimingEventProps			
Aggregated by	CompositionSwComponentType.instantiationRTEEventProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***InstantiationRTEEventProps*** (abstract)			
refinedEvent	RTEEvent	0..1	iref	This instance ref denotes the Timing Event for which the period shall be refined on an instance level. **Tags:** atp.Splitkey=refinedEvent, refinedEvent.context ComponentPrototype, refinedEvent.targetEvent **InstanceRef implemented by:** InstanceEventIn CompositionInstanceRef
shortLabel	Identifier	0..1	attr	The main purpose of the shortLabel is to contribute to the splitability. **Stereotypes:** atpIdentityContributor

Table A.641: InstantiationRTEEventProps

Class	**InstantiationTimingEventProps**			
Note	This meta-class represents the ability to refine a timing event for particular instances of a software component. This approach supports an instance specific timing.			
Base	*ARObject*, *InstantiationRTEEventProps*			
Aggregated by	CompositionSwComponentType.instantiationRTEEventProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
period	TimeValue	0..1	attr	This attribute represents the value of the refined activation period.

Table A.642: InstantiationTimingEventProps

Class	***InternalBehavior*** (abstract)			
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			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
constant Memory	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior. The shortName of ParameterDataPrototype has to be equal to the ''C' identifier of the described constant. The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=constantMemory, constantMemory.short Name, constantMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior **Tags:** atp.Splitkey=constantValueMapping

▽

△

Class	***InternalBehavior*** (abstract)			
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior **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:** atpVariation **Tags:** atp.Splitkey=exclusiveArea, exclusiveArea.shortName, exclusiveArea.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaNestingOrder, exclusiveArea NestingOrder.shortName, exclusiveAreaNesting Order.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:** atpVariation **Tags:** atp.Splitkey=staticMemory, staticMemory.shortName, staticMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.643: InternalBehavior

Class	**InternalConstrs**			
Note	This meta-class represents the ability to express internal constraints.			
Base	*ARObject*			
Aggregated by	DataConstrRule.internalConstrs			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

▽

△

Class	InternalConstrs			
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 maps and curves. **Tags:** xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This element specifies the monotony characteristics of the current internal or physical limits. The following table shows the monotony characteristics which are to be filled through the corresponding values. If the element has no contents or if it is omitted, "no Monotony" is the default content. **Tags:** xml.sequenceOffset=70
upperLimit	Limit	0..1	attr	This specifies the upper limit defined by the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table A.644: InternalConstrs

Class	InternalTriggerOccurredEvent			
Note	This event is raised when the referenced `InternalTriggeringPoint` has occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	InternalTriggeringPoint	0..1	ref	The referenced `InternalTriggeringPoint` raises this `InternalTriggerOccurredEvent`. **Tags:** atp.Splitkey=eventSource

Table A.645: InternalTriggerOccurredEvent

Class	InternalTriggeringPoint			
Note	If a `RunnableEntity` owns an `InternalTriggeringPoint` it is entitled to trigger the execution of `RunnableEntity`s of the corresponding software-component.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.internalTriggeringPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers.

Table A.646: InternalTriggeringPoint

Class	InterpolationRoutine
Note	This represents an interpolation routine taken to evaluate the contents of a curve or map against a specific input value.
Base	*ARObject*

▽

△

Class	**InterpolationRoutine**			
Aggregated by	InterpolationRoutineMapping.interpolationRoutine			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interpolation Routine	BswModuleEntry	0..1	ref	This specifies a BswModuleEntry which implements the current interpolation method for the given record layout. **Tags:** atp.Splitkey=interpolationRoutine xml.sequenceOffset=30 This Attribute is only used by the AUTOSAR Classic Platform.
isDefault	Boolean	0..1	attr	This attribute specifies whether the enclosing InterpolationRoutine is considered the default in the context (defined by the System Template) of a given collection InterpolationRoutineMapping that owns the enclosing InterpolationRoutine. **Tags:** xml.sequenceOffset=20
shortLabel	Identifier	0..1	attr	This is the name of the interpolation method which is implemented by the referenced bswModuleEntry. It corresponds to swInterpolationMethod in SwDataDef Props. **Tags:** xml.sequenceOffset=10

Table A.647: InterpolationRoutine

Class	**InterpolationRoutineMapping**			
Note	This meta-class provides a mapping between one record layout and its matching interpolation routines. This allows to formally specify the semantics of the interpolation routines. The use case is such that the curves/Maps define an interpolation method. This mapping table specifies which interpolation routine implements methods for a particular record layout. Using this information, the implementer of a software-component can select the appropriate interpolation routine.			
Base	*ARObject*			
Aggregated by	InterpolationRoutineMappingSet.interpolationRoutineMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interpolation Routine	InterpolationRoutine	*	aggr	This is one particular interpolation routine which is mapped to the record layout. **Tags:** atp.Splitkey=interpolationRoutine
swRecord Layout	SwRecordLayout	0..1	ref	This refers to the record layout which is mapped to interpolation routines. **Tags:** atp.Splitkey=swRecordLayout

Table A.648: InterpolationRoutineMapping

Class	**InterpolationRoutineMappingSet**			
Note	This meta-class specifies a set of interpolation routine mappings. **Tags:** atp.recommendedPackage=InterpolationRoutineMappingSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**InterpolationRoutineMappingSet**			
interpolation Routine Mapping	InterpolationRoute Mapping	*	aggr	This specifies one particular mapping of recordlayout and its matching interpolationRoutines. **Tags:** atp.Splitkey=interpolationRoutineMapping

Table A.649: InterpolationRoutineMappingSet

Class	**InvalidationPolicy**			
Note	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`.			
Base	*ARObject*			
Aggregated by	SenderReceiverInterface.invalidationPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	0..1	ref	Reference to the `dataElement` for which the `InvalidationPolicy` applies. **Tags:** atp.Splitkey=dataElement
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute controls how invalidation is applied to the dataElement.

Table A.650: InvalidationPolicy

Class	**Ipv4ArpProps**			
Note	Specifies the configuration options for the ARP (Address Resolution Protocol).			
Base	*ARObject*			
Aggregated by	Ipv4Props.arpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpArpNum GratuitousArp OnStartup	PositiveInteger	0..1	attr	This attribute specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.
tcpIpArpPacket QueueEnabled	Boolean	0..1	attr	This attribute enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.
tcpIpArp Request Timeout	TimeValue	0..1	attr	This attribute specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the TcpIp shall skip the transmission of any further ARP requests to the same destination within a duration of tcpIpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1).
tcpIpArpTable EntryTimeout	TimeValue	0..1	attr	This attribute specifies the timeout in seconds after which an unused ARP entry is removed.

Table A.651: Ipv4ArpProps

Class	**Ipv4Configuration**			
Note	Internet Protocol version 4 (IPv4) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**Ipv4Configuration**			
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.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
ipAddressKeep Behavior	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. **Stereotypes:** atpIdentityContributor
ipv4Address Source	Ipv4AddressSource Enum	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 A.652: Ipv4Configuration

Class	**Ipv4FragmentationProps**			
Note	Specifies the configuration options for IPv4 packet fragmentation/reassembly.			
Base	*ARObject*			
Aggregated by	Ipv4Props.fragmentationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpIp Fragmentation RxEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).
tcpIpIpNum Fragments	PositiveInteger	0..1	attr	Specifies the maximum number of IP fragments per datagram.
tcpIpIpNum ReassDgrams	PositiveInteger	0..1	attr	Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel.
tcpIpIpReass Timeout	TimeValue	0..1	attr	Specifies the timeout in [s] after which an incomplete datagram gets discarded.

Table A.653: Ipv4FragmentationProps

Class	**Ipv6Configuration**			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.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.

▽

△

Class	**Ipv6Configuration**			
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. **Stereotypes:** atpIdentityContributor
ipv6Address Source	Ipv6AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.

Table A.654: Ipv6Configuration

Class	**Ipv6FragmentationProps**			
Note	This meta-class specifies the configuration options for IPv6 packet fragmentation/reassembly.			
Base	*ARObject*			
Aggregated by	Ipv6Props.fragmentationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpIp Reassembly BufferCount	PositiveInteger	0..1	attr	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. A value of 0 disables fragment reassembly.
tcpIpIp Reassembly BufferSize	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.
tcpIpIp Reassembly SegmentCount	PositiveInteger	0..1	attr	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. To deal with fragments received out of order this value should be configured bigger than 1.
tcpIpIp Reassembly Timeout	TimeValue	0..1	attr	Specifies the timeout in seconds after which an incomplete datagram gets discarded.
tcpIpIpTx FragmentBuffer Count	PositiveInteger	0..1	attr	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.
tcpIpIpTx FragmentBuffer Size	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.

Table A.655: Ipv6FragmentationProps

Class	**Ipv6NdpProps**			
Note	This meta-class specifies the configuration options for the Neighbor Discovery Protocol for IPv6.			
Base	*ARObject*			
Aggregated by	Ipv6Props.ndpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpNdpDefault ReachableTime	TimeValue	0..1	attr	Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RetransTimer	TimeValue	0..1	attr	Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RouterListSize	PositiveInteger	0..1	attr	Maximum number of default router entries.
tcpIpNdp Defensive Processing	Boolean	0..1	attr	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.
tcpIpNdpDelay FirstProbeTime Value	TimeValue	0..1	attr	Delay before sending the first NUD probe in (s).
tcpIpNdp Destination CacheSize	PositiveInteger	0..1	attr	Maximum number of entries in the destination cache.
tcpIpNdp DynamicHop LimitEnabled	Boolean	0..1	attr	If enabled the default hop limit may be reconfigured based on received Router Advertisements.
tcpIpNdp DynamicMtu Enabled	Boolean	0..1	attr	Allow dynamic reconfiguration of link MTU via Router Advertisements.
tcpIpNdp Dynamic ReachableTime Enabled	Boolean	0..1	attr	If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.
tcpIpNdp Dynamic RetransTime Enabled	Boolean	0..1	attr	If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.
tcpIpNdpMax RandomFactor	PositiveInteger	0..1	attr	Maximum random factor used for randomization
tcpIpNdpMaxRtr Solicitation Delay	TimeValue	0..1	attr	Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).
tcpIpNdpMaxRtr Solicitations	PositiveInteger	0..1	attr	Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.
tcpIpNdpMin RandomFactor	PositiveInteger	0..1	attr	Minimum random factor used for randomization
tcpIpNdp Neighbor Unreachability Detection Enabled	Boolean	0..1	attr	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.
tcpIpNdpNum Multicast Solicitations	PositiveInteger	0..1	attr	Maximum number of multicast solicitations that will be sent when performing address resolution.
tcpIpNdpNum Unicast Solicitations	PositiveInteger	0..1	attr	Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection.



△

Class	**Ipv6NdpProps**			
tcpIpNdpPacket QueueEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.
tcpIpNdpPrefix ListSize	PositiveInteger	0..1	attr	Maximum number of entries in the on-link prefix list.
tcpIpNdp Random ReachableTime Enabled	Boolean	0..1	attr	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.
tcpIpNdpRndRtr Solicitation DelayEnabled	Boolean	0..1	attr	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.
tcpIpNdpRtr Solicitation Interval	TimeValue	0..1	attr	Interval between consecutive Router Solicitations in (s).
tcpIpNdpSlaac DadNumberOf Transmissions	PositiveInteger	0..1	attr	Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.
tcpIpNdpSlaac Dad Retransmission Delay	TimeValue	0..1	attr	Sets the maximum value for the address configuration delay (s).
tcpIpNdpSlaac DelayEnabled	Boolean	0..1	attr	If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].
tcpIpNdpSlaac OptimisticDad Enabled	Boolean	0..1	attr	Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.

Table A.656: Ipv6NdpProps

Class	«atpVariation» **J1939Cluster**			
Note	J1939 specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *AbstractCanCluster*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dcmBusType	J1939BusTypeEnum	0..1	attr	Identification of a network to which the broadcast of messages is started by the DM13 command message.
request2 Support	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.

▽

△

Class	«atpVariation» **J1939Cluster**			
usesMultiPg	Boolean	0..1	attr	This attribute shall only be configured for a CanCluster that supports CAN FD. If configured and set to False, the J1939 communication on this cluster is expected to adhere to J1939-21. Otherwise, the J1939 communication is expected to adhere to J1939-22. Note: The communication of the J1939 specific modules is derived taking this setting into account.

Table A.657: J1939Cluster

Class	**J1939ControllerApplication**			
Note	This element represents a J1939 controller application. **Tags:** atp.recommendedPackage=J1939ControllerApplications This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
functionId	PositiveInteger	0..1	attr	This attribute represents the numerical function id of the J1939 controller application.
swComponent Prototype	SwComponent Prototype	0..1	iref	This represents the SwComponentPrototype (which is typically typed by a CompositionSwComponentType) that corresponds to the J1939ControllerApplication. **Tags:** atp.Splitkey=swComponentPrototype, swComponentPrototype.contextComposition, swComponentPrototype.contextComponent, swComponentPrototype.targetComponent **InstanceRef implemented by:** ComponentInSystemInstanceRef

Table A.658: J1939ControllerApplication

Class	**J1939ControllerApplicationToJ1939NodeMapping**			
Note	This meta-class represents the ability to map a J1939ControllerApplication to a J1939Node. 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 J1939Node. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	SystemMapping.j1939ControllerApplicationToJ1939NodeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
j1939Controller Application	J1939Controller Application	0..1	ref	Reference to the J1939 Controller Application that is mapped to the referenced J1939Node. **Tags:** atp.Splitkey=j1939ControllerApplication
j1939Node	J1939Node	0..1	ref	J1939Node that is the target of the J1939Controller ApplicationTo1939NodeMapping. **Tags:** atp.Splitkey=j1939Node

Table A.659: J1939ControllerApplicationToJ1939NodeMapping

Class	J1939DcmIPdu			
Note	Represents the IPdus handled by J1939Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic MessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table A.660: J1939DcmIPdu

Class	J1939NmCluster			
Note	J1939 specific NmCluster attributes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
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 nmNodeId might still be necessary for the J1939Rm and J1939Tp.
usesDynamic Addressing	Boolean	0..1	attr	Defines whether fully dynamic address resolution according to SAE J1939-81 shall be supported on this J1939NmCluster. • True: The dynamically allocated addresses on the bus are matched at runtime to the configured addresses. • False: The addresses on the bus resemble the configured addresses.

Table A.661: J1939NmCluster

Class	J1939NmNode			
Note	J1939 specific NM Node attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
address Configuration Capability	J1939NmAddress ConfigurationCapability Enum	0..1	attr	Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).
j1939Node	J1939Node	0..1	ref	Reference to J1939Node that has a defined J1939Node Name and is assocated with a J1939Controller Application. **Tags:** atp.Splitkey=j1939Node

Table A.662: J1939NmNode

Class	J1939Node			
Note	J1939Node defined on the aggregating EcuInstance that is associated with J1939ControllerApplication via a J1939ControllerApplicationTo1939NodeMapping. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.j1939Node			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nodeName	J1939NodeName	0..1	aggr	Collectzion of attributes to configure the NodeName. **Tags:** atp.Splitkey=nodeName
pduTriggering	PduTriggering	*	ref	Reference to Pdus that are related to the J1939Node and in turn to a J1939ControllerApplication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.663: J1939Node

Class	J1939NodeName			
Note	This element contains attributes to configure the J1939NmNode NAME. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	J1939Node.nodeName			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arbitrary Address Capable	Boolean	0..1	attr	Arbitrary Address Capable field of the NAME of this node.
ecuInstance	Integer	0..1	attr	ECU Instance field of the NAME of this node.
function	Integer	0..1	attr	Function field of the NAME of this node.
function Instance	Integer	0..1	attr	Function Instance field of the NAME of this node.
identitiyNumber	Integer	0..1	attr	Identity Number field of the NAME of this node.
industryGroup	Integer	0..1	attr	Industry Group field of the NAME of this node.
manufacturer Code	Integer	0..1	attr	Manufacturer Code field of the NAME of this node.
vehicleSystem	Integer	0..1	attr	Vehicle System field of the NAME of this node.
vehicleSystem Instance	Integer	0..1	attr	Vehicle System Instance field of the NAME of this node.

Table A.664: J1939NodeName

Class	J1939ProtectedIPdu			
Note	Represents the SDM (Safety Data Message) that contains the actual payload (signals) of the E2E protected J1939 message. **Tags:** atp.Status=draft atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**J1939ProtectedIPdu**			
payload	PduTriggering	0..1	ref	References the ISignalIPdu that represents the SDG (Safety Data Group) that contains both the payload (signals) and the E2E protection data of the protected J1939 message. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=payload, payload.pduTriggering, payload.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
srvt	TimeValue	0..1	attr	Maximum time between SHM (Safety Header Message) and SDM (Safety Data Message) of one SDG (Safety Data Group) **Tags:** atp.Status=draft

Table A.665: J1939ProtectedIPdu

Class	**J1939TpConfig**			
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 This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpAddress	TpAddress	*	aggr	Collection of TP Adresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	J1939TpConnection	*	aggr	Configuration of J1939 TP connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	J1939TpNode	*	aggr	Senders and receivers of J1939 TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.666: J1939TpConfig

Class	**J1939TpConnection**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	J1939TpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
acceptVariableDA	Boolean	0..1	attr	The TP message is accepted independently of the actually used destination address (DA). Otherwise, only the destination address configured as receiver.tpAddress is accepted. Only derived for the receiving ECU.
acceptVariableSA	Boolean	0..1	attr	The TP message is accepted independently of the actually used source address (SA). Otherwise, only the source address configured as transmitter.tpAddress is accepted. Only derived for the receiving ECU.
bamTiming	TimeValue	0..1	attr	Timing of TP.DT frames of a BAM transmission. Range 10ms to 50ms, if not configured 50ms are assumed. Relevant for tpProtocolTypes bam, bam_cmdt, fd_bam, and fd_bam_cmdt.
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	0..1	ref	Data Message (TP.DT) used by CMDT and BAM. The DataNPdu has a fixed length of 8 bytes. **Tags:** atp.Splitkey=dataPdu
dynamicBs	Boolean	0..1	attr	Enable support for dynamic block size calculation.
flowControlPdu	NPdu	0..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 compatibilty reasons. **Tags:** atp.Splitkey=flowControlPdu
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. **Tags:** atp.Splitkey=receiver
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. **Tags:** atp.Splitkey=tpPg
tpProtocolType	J1939TpProtocolTypeEnum	0..1	attr	Protocol type used by the J1939TpConnection
transmitter	J1939TpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter
txPduPriority	PositiveInteger	0..1	attr	If configured, TP.CM and TP.DT frames shall be sent with this priority instead of the default priority 7. Range 0 to 7.



△

Class	**J1939TpConnection**			
useVariableDA	Boolean	0..1	attr	The TP message is sent with variable destination address (DA). Otherwise, the destination address configured as receiver.tpAddress is always used. Only derived for the transmitting ECU.
useVariableSA	Boolean	0..1	attr	The TP message is sent with variable source address (SA). Otherwise, the source address configured as transmitter.tpAddress is always used. Only derived for the transmitting ECU.

Table A.667: J1939TpConnection

Class	**J1939TpNode**			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	J1939TpConfig.tpNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connector	Communication Connector	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). **Tags:** atp.Splitkey=connector
j1939Node	J1939Node	0..1	ref	Reference to J1939Node that has a defined J1939Node Name and is assocated with a J1939Controller Application. **Tags:** atp.Splitkey=j1939Node
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. **Tags:** atp.Splitkey=tpAddress

Table A.668: J1939TpNode

Class	**J1939TpPg**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	J1939TpConnection.tpPg			
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. **Tags:** atp.Splitkey=directPdu
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.

▽

△

Class	**J1939TpPg**			
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, CMDT, direct) on the same J1939TpConnection. **Tags:** atp.Splitkey=sdu
txDirectPdu Priority	PositiveInteger	0..1	attr	If configured, direct frames shall be sent with this priority instead of the priority configured in the CAN ID.

Table A.669: J1939TpPg

Enumeration	**J1939TpProtocolTypeEnum**
Note	Protocol types that may be used by the J1939TpConnection
Aggregated by	J1939TpConnection.tpProtocolType
Literal	*Description*
bam	Use BAM protocol variant of J1939-21 exclusively. **Tags:** atp.EnumerationLiteralIndex=0
bam_cmdt	Use BAM or CMDT, depending on Meta Data for the destination address. **Tags:** atp.EnumerationLiteralIndex=2
cmdt	Use CMDT protocol variant of J1939-21 (aka RTS/CTS) exclusively. **Tags:** atp.EnumerationLiteralIndex=1
etp	Use ETP protocol of ISO 11783-6. **Tags:** atp.EnumerationLiteralIndex=6
fd_bam	Use FD BAM protocol variant of J1939-22 exclusively. **Tags:** atp.EnumerationLiteralIndex=3
fd_bam_cmdt	Use FD BAM or FD CMDT, depending on Meta Data for the destination address. **Tags:** atp.EnumerationLiteralIndex=5
fd_cmdt	Use FD CMDT protocol variant of J1939-22 (aka FD RTS/CTS) exclusively. **Tags:** atp.EnumerationLiteralIndex=4
fpp	Use fast packet protocol of NMEA 2000. **Tags:** atp.EnumerationLiteralIndex=7

Table A.670: J1939TpProtocolTypeEnum

Class	«atpMixedString» **LParagraph**			
Note	This is the text for a paragraph in one particular language. The language is denoted in the attribute l.			
Base	*ARObject*, *LanguageSpecific*, *MixedContentForParagraph*			
Aggregated by	MultiLanguageParagraph.l1			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.671: LParagraph

Enumeration	**LatencyConstraintTypeEnum**
Note	Specifies the `latencyConstraintType` for a `LatencyTimingConstraint`.
Aggregated by	LatencyTimingConstraint.latencyConstraintType
Literal	***Description***
age	The `LatencyTimingConstraint` is seen from the perspective of the `response` event of the `scope`. Given a certain `response` event, the age interval of the latest `stimulus` is constrained. **Tags:** atp.EnumerationLiteralIndex=0
reaction	The `LatencyTimingConstraint` is seen from the perspective of the `stimulus` event of the `scope`. Given a certain `stimulus` event, the reaction interval of the first `response` is constrained. **Tags:** atp.EnumerationLiteralIndex=1

Table A.672: LatencyConstraintTypeEnum

Class	**LatencyTimingConstraint**			
Note	Constrains the time duration between the occurrence of the `stimulus` and the occurrence of the corresponding `response` of that `scope`. A causal dependency between the `stimulus` and the corresponding `response` of the `scope` is required.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
latency ConstraintType	LatencyConstraintType Enum	0..1	attr	The specific type of this latency constraint.
maximum	MultidimensionalTime	0..1	aggr	The maximum latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain, i.e. `scope` should not exceed this upper bound **Tags:** atp.Splitkey=maximum xml.sequenceOffset=20
minimum	MultidimensionalTime	0..1	aggr	The minimum latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain, i.e. `scope` should not precede this lower bound **Tags:** atp.Splitkey=minimum xml.sequenceOffset=10
nominal	MultidimensionalTime	0..1	aggr	The nominal latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain. **Tags:** atp.Splitkey=nominal xml.sequenceOffset=30
scope	TimingDescriptionEvent Chain	0..1	ref	The event chain that defines the scope of the constraint. **Tags:** atp.Splitkey=scope

Table A.673: LatencyTimingConstraint

Class	**LifeCycleInfo**
Note	LifeCycleInfo describes the life cycle state of an element together with additional information like what to use instead
Base	*ARObject*
Aggregated by	LifeCycleInfoSet.lifeCycleInfo

▽

△

Class	**LifeCycleInfo**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lcObject	Referrable	0..1	ref	Element(s) have the life cycle as described in lcState. **Tags:** atp.Splitkey=lcObject
lcState	LifeCycleState	0..1	ref	This denotes the particular state assigned to the object. If no lcState is given then the default life cycle state of Life CycleInfoSet is assumed. **Tags:** atp.Splitkey=lcState
periodBegin	LifeCyclePeriod	0..1	aggr	Starting point of period in which the element has the denoted life cycle state lcState. If no periodBegin is given then the default period begin of LifeCycleInfoSet is assumed. **Tags:** atp.Splitkey=periodBegin
periodEnd	LifeCyclePeriod	0..1	aggr	Expiry date, i.e. end point of period the element does not have the denoted life cycle state lcState any more. If no periodEnd is given then the default period begin of Life CycleInfoSet is assumed. **Tags:** atp.Splitkey=periodEnd
remark	DocumentationBlock	0..1	aggr	Remark describing for example • why the element was given the specified life cycle • the semantics of useInstead **Tags:** atp.Splitkey=remark
useInstead	Referrable	*	ref	Element(s) that should be used instead of the one denoted in referrable. Only relevant in case of life cycle states lcState unlike “valid”. In case there are multiple references the exact semantics shall be individually described in the remark. **Tags:** atp.Splitkey=useInstead

Table A.674: LifeCycleInfo

Class	**LifeCycleInfoSet**			
Note	This meta class represents the ability to attach a life cycle information to a particular set of elements. The information can be defined for a particular period. This supports the definition of transition plans. If no period is specified, the life cycle state applies forever. **Tags:** atp.recommendedPackage=LifeCycleInfoSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultLcState	LifeCycleState	0..1	ref	This denotes the default life cycle state. To be used in all LifeCycleInfo elements within the LifeCycleInfoSet if no life cycle state is stated there explicitly. I.e. the defaultLc State can be overwritten in LifeCycleInfo elements. **Tags:** atp.Splitkey=defaultLcState
defaultPeriod Begin	LifeCyclePeriod	0..1	aggr	Default starting point of period in which all the specified lifeCycleInfo apply. Note that the default period can be overridden for each lifeCycleInfo individually. **Tags:** atp.Splitkey=defaultPeriodBegin

▽

△

Class	**LifeCycleInfoSet**			
defaultPeriod End	LifeCyclePeriod	0..1	aggr	Default expiry date, i.e. default end point of period for which all specified lifeCycleInfo apply. Note that the default period can be overridden for each lifeCycleInfo individually. **Tags:** atp.Splitkey=defaultPeriodEnd
lifeCycleInfo	LifeCycleInfo	*	aggr	This represents one particular life cycle information. **Tags:** atp.Splitkey=lifeCycleInfo
usedLifeCycle StateDefinition Group	LifeCycleStateDefinition Group	0..1	ref	This denotes the life cycle states applicable to the current life cycle info set. **Tags:** atp.Splitkey=usedLifeCycleStateDefinitionGroup

Table A.675: LifeCycleInfoSet

Class	**LifeCyclePeriod**			
Note	This meta class represents the ability to specify a point of time within a specified period, e.g. the starting or end point, in which a specific life cycle state is valid/applies to.			
Base	*ARObject*			
Aggregated by	LifeCycleInfo.periodBegin, LifeCycleInfo.periodEnd, LifeCycleInfoSet.defaultPeriodBegin, LifeCycleInfo Set.defaultPeriodEnd			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arRelease Version	RevisionLabelString	0..1	attr	Version of the AUTOSAR Release the element referred to is part of. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR. **Tags:** xml.sequenceOffset=20
date	DateTime	0..1	attr	Date within period. **Tags:** xml.sequenceOffset=10
productRelease	RevisionLabelString	0..1	attr	Version of the product within the period. **Tags:** xml.sequenceOffset=30

Table A.676: LifeCyclePeriod

Class	**LifeCycleState**			
Note	This meta class represents one particular state in the LifeCycle.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LifeCycleStateDefinitionGroup.lcState			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.677: LifeCycleState

Class	**LifeCycleStateDefinitionGroup**
Note	This meta class represents the ability to define the states and properties of one particular life cycle. **Tags:** atp.recommendedPackage=LifeCycleStateDefinitionGroups
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**LifeCycleStateDefinitionGroup**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lcState	LifeCycleState	*	aggr	Describes a single life cycle state of this life cycle state definition group. **Tags:** atp.Splitkey=lcState, lcState.shortName

Table A.678: LifeCycleStateDefinitionGroup

Primitive	**Limit**			
Note	This class represents the ability to express a numerical limit. Note that this is in fact a NumericalVariation Point but has the additional attribute intervalType. **Tags:** xml.xsd.customType=LIMIT-VALUE xml.xsd.pattern=(0[xX][0-9a-fA-F]+)|(0[0-7]+)|(0[bB][0-1]+)|(([+\-]?[1-9][0-9]+(\.[0-9]+)?|[+\-]?[0-9](\.[0-9]+)?)([eE]([+\-]?)[0-9]+)?)|\.0|INF|-INF|NaN xml.xsd.type=string			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
intervalType	IntervalTypeEnum	0..1	attr	This specifies the type of the interval. If the attribute is missing the interval shall be considered as "CLOSED". **Tags:** xml.attribute=true

Table A.679: Limit

Class	«atpVariation» **LinCluster**			
Note	LIN specific attributes **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.680: LinCluster

Class	**LinCommunicationConnector**			
Note	LIN bus specific communication connector attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurable Frame	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. **Tags:** atp.Splitkey=linConfigurableFrame

▽

△

Class	**LinCommunicationConnector**			
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	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. **Tags:** atp.Splitkey=linOrderedConfigurableFrame
schedule ChangeNext TimeBase	Boolean	0..1	attr	This attribute defines the point in time where a schedule table switch is performed. If this attribute is set to false or not present, the schedule table shall be switched after the current entry of the active schedule table is ended. If this attribute is enabled, the schedule table shall be switched when message transmission or reception within an entry has been completed, ensured by status checks for transmission and reception.

Table A.681: LinCommunicationConnector

Class	«atpVariation» ***LinCommunicationController*** (abstract)			
Note	LIN bus specific communication controller attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	LinMaster, LinSlave			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
protocolVersion	String	0..1	attr	Version specifier for a communication protocol.

Table A.682: LinCommunicationController

Class	**LinConfigurableFrame**			
Note	Assignment of messageIds to Frames. This element shall be used for the LIN 2.0 Assign-Frame command. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinCommunicationConnector.linConfigurableFrame, LinSlaveConfig.linConfigurableFrame			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
frame	LinFrame	0..1	ref	Reference to a Frame that is processed by the slave node. **Tags:** atp.Splitkey=frame
messageId	PositiveInteger	0..1	attr	MessageId for the referenced frame

Table A.683: LinConfigurableFrame

Class	**LinErrorResponse**
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. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*

▽

△

Class	**LinErrorResponse**			
Aggregated by	LinSlave.linErrorResponse, LinSlaveConfig.linErrorResponse			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
responseError	ISignalTriggering	0..1	ref	This ISignal shall be taken to transport the responseError bit. **Tags:** atp.Splitkey=responseError

Table A.684: LinErrorResponse

Class	**LinEventTriggeredFrame**			
Note	An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response. 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. The event controlled frame shall not contain any Pdus. **Tags:** atp.recommendedPackage=Frames This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
collision Resolving Schedule	LinScheduleTable	0..1	ref	Reference to the schedule table, which resolves a collision. **Tags:** atp.Splitkey=collisionResolvingSchedule
linUnconditional Frame	LinUnconditionalFrame	*	ref	A list of slaves can respond to the master request if at least one of the signals carried in its unconditional frame is updated. 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. The Unconditional frames associated with an event triggered frame 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 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. **Tags:** atp.Splitkey=linUnconditionalFrame

Table A.685: LinEventTriggeredFrame

Class	***LinFrame*** (abstract)			
Note	Lin specific Frame element. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	LinEventTriggeredFrame, LinSporadicFrame, LinUnconditionalFrame			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.686: LinFrame

Class	**LinFrameTriggering**			
Note	LIN specific attributes to the FrameTriggering This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
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 A.687: LinFrameTriggering

Class	«atpVariation» **LinMaster**			
Note	Describing the properties of the refering ecu as a LIN master. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *LinCommunicationController*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
linSlave	LinSlaveConfig	*	aggr	LinSlaves that are handled by the LinMaster. **Tags:** atp.Splitkey=linSlave, linSlave.ident.shortName
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 A.688: LinMaster

Class	LinOrderedConfigurableFrame			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinCommunicationConnector.linOrderedConfigurableFrame, LinSlaveConfig.linOrderedConfigurableFrame			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frame	LinFrame	0..1	ref	Reference to a Frame that is processed by the slave node. **Tags:** atp.Splitkey=frame
index	Integer	0..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 A.689: LinOrderedConfigurableFrame

Class	LinPhysicalChannel			
Note	LIN specific attributes to the physicalChannel This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
busIdleTimeout Period	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:** atp.Splitkey=scheduleTable, scheduleTable.shortName, scheduleTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.690: LinPhysicalChannel

Class	LinScheduleTable			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LinPhysicalChannel.scheduleTable			
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.



△

Class	**LinScheduleTable**			
tableEntry	ScheduleTableEntry	*	aggr	The scheduling table consists of table entries, which contain Frame slots. **Tags:** atp.Splitkey=tableEntry

Table A.691: LinScheduleTable

Class	«atpVariation» **LinSlave**			
Note	Describing the properties of the referring ecu as a LIN slave. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *LinCommunicationController*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
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	0..1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	0..1	attr	LIN function ID
initialNad	Integer	0..1	attr	This attribute represents the initial NAD.
linError Response	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames. **Tags:** atp.Splitkey=linErrorResponse
nasTimeout	TimeValue	0..1	attr	Value of the N_AS timeout. Unit: seconds.
supplierId	PositiveInteger	0..1	attr	LIN Supplier ID
variantId	PositiveInteger	0..1	attr	Specifies the Variant ID

Table A.692: LinSlave

Class	**LinSlaveConfig**			
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). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinMaster.linSlave			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
configuredNad	Integer	0..1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	0..1	attr	LIN function ID.
ident	LinSlaveConfigIdent	0..1	aggr	This adds the ability to become referrable to LinSlave Config. **Stereotypes:** atpIdentityContributor
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.

▽

△

Class	LinSlaveConfig			
linConfigurable Frame	LinConfigurableFrame	*	aggr	List of all frames that are processed by the slave node **Tags:** atp.Splitkey=linConfigurableFrame
linError Response	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames. **Tags:** atp.Splitkey=linErrorResponse
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	aggr	List of 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. **Tags:** atp.Splitkey=linOrderedConfigurableFrame
protocolVersion	String	0..1	attr	Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different.
supplierId	PositiveInteger	0..1	attr	LIN Supplier ID.
variantId	PositiveInteger	0..1	attr	Specifies the Variant ID.

Table A.693: LinSlaveConfig

Class	LinSporadicFrame			
Note	A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus. **Tags:** atp.recommendedPackage=Frames This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
substituted Frame (ordered)	LinUnconditionalFrame	*	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 A.694: LinSporadicFrame

Class	LinTpConfig
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 This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*
Aggregated by	ARPackage.element

▽

△

Class	**LinTpConfig**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpAddress	TpAddress	*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	LinTpConnection	*	aggr	Configuration of LIN TP channels. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	LinTpNode	*	aggr	Senders and receivers of LIN TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.695: LinTpConfig

Class	**LinTpConnection**			
Note	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. 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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	LinTpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataPdu	NPdu	0..1	ref	Reference to an NPdu (Single Frame, First Frame or Consecutive Frame). 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. 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. The Consecutive Frame network protocol data unit (CF N_PDU) transfers segments (N_Data) of the service data unit message data (<MessageData>). All network ▽

▽

△

Class	LinTpConnection			
				△ protocol data units (N_PDUs) 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_PDUs). **Tags:** atp.Splitkey=dataPdu
flowControl	NPdu	0..1	ref	Reference to the Flow Control NPdu. 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_PDUs. 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: a) First Frame network protocol data unit (FF N_PDU) 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. **Tags:** atp.Splitkey=flowControl
linTpNSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=linTpNSdu
multicast	TpAddress	0..1	ref	TP address for 1:n connections. **Tags:** atp.Splitkey=multicast
receiver	LinTpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver
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	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter

Table A.696: LinTpConnection

Class	LinTpNode			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LinTpConfig.tpNode			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	LinTpNode			
connector	Communication Connector	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). **Tags:** atp.Splitkey=connector
dropNot RequestedNad	Boolean	0..1	attr	Configures if TP Frames of not requested LIN-Slaves are dropped or not.
maxNumberOf RespPending Frames	Integer	0..1	attr	Configures the maximum number of allowed response pending frames.
p2Max	TimeValue	0..1	attr	After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.
p2Timing	TimeValue	0..1	attr	P2 timeout observation parameter.
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). **Tags:** atp.Splitkey=tpAddress

Table A.697: LinTpNode

Class	LinUnconditionalFrame			
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 This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.698: LinUnconditionalFrame

Class	MacAddressVlanMembership			
Note	Assigns a set of MAC-Multicast-Addresses which are addressable via the CouplingPort aggregating this MacAddressVlanMembership. Optionally also assigns a set of VLANs to this relation. This is a static pre-configuration and further addresses may be learned during runtime.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPort.macAddressVlanAssignment			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
macMulticast Address	MacMulticastGroup	*	ref	Defines a set of macMulticastAddresses to be mapped to the CouplingPort. **Tags:** atp.Splitkey=macMulticastAddress
vlan	EthernetPhysical Channel	*	ref	Defines a set of VLANs the set of macMulticastAddress apply to. **Tags:** atp.Splitkey=vlan

Table A.699: MacAddressVlanMembership

Class	MacMulticastConfiguration			
Note	References a per cluster globally defined MAC-Multicast-Group.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
macMulticast Group	MacMulticastGroup	0..1	ref	Reference to a macMulticastGroup. **Stereotypes:** atpIdentityContributor

Table A.700: MacMulticastConfiguration

Class	MacMulticastGroup			
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	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetCluster.macMulticastGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
macMulticast Address	MacAddressString	0..1	attr	A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.

Table A.701: MacMulticastGroup

Class	MacSecBypassedCommunication			
Note	Definition of bypassed communication for MACsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *AbstractMacSecCommunication*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.702: MacSecBypassedCommunication

Class	MacSecConnectivityAssociation			
Note	Definition of a Connectivity Association for MACsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *AbstractMacSecCommunication*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.703: MacSecConnectivityAssociation

Class	MacSecProps
Note	This meta-class allows to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:** atp.Status=candidate
Base	*ARObject*

▽

△

Class	**MacSecProps**			
Aggregated by	CouplingPort.macSecProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
autoStart	Boolean	0..1	attr	This attribute defines how the Port Access Entity (PAE) is started: • true := Autostart • false := Manual Start **Tags:** atp.Status=candidate
macSecKay Config	MacSecLocalKayProps	0..1	aggr	Properties to configure the MKA instance (KaY) for a controlled CouplingPort (PaE). **Tags:** atp.Splitkey=macSecKayConfig atp.Status=candidate
onFail Permissive Mode	MacSecFailPermissive ModeEnum	0..1	attr	This attribute sets the behavior of the Port Access Entity in case MACsec does not succeed. **Tags:** atp.Status=candidate
onFail Permissive ModeTimeout	TimeValue	0..1	attr	Timeout in seconds to enable the controlled port in case onFailPermissiveMode is set to Timeout. **Tags:** atp.Status=candidate
sakRekeyTime Span	TimeValue	0..1	attr	Time in seconds to trigger the rekey of an in use SAK (Static Secure Association key). If set to 0, the rekey will not be triggered after a time span. **Tags:** atp.Status=candidate

Table A.704: MacSecProps

Class	**McDataAccessDetails**			
Note	This meta-class allows to attach detailed information about the usage of a data buffer by the RTE to a corresponding McDataInstance. Use Case: Direct memory access to RTE internal buffers for rapid prototyping. In case of implicit communication, the various task local buffers need to be identified in relation to RTE events and variable access points. Note that the SwComponentPrototype, the RunnableEntity and the VariableDataPrototype are implicitly given be the referred instances of RTEEvent and VariableAccess.			
Base	*ARObject*			
Aggregated by	McDataInstance.mcDataAccessDetails			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
rteEvent	RTEEvent	*	iref	The RTE event used to receive the data via this buffer. **Tags:** atp.Splitkey=rteEvent, rteEvent.contextRoot Composition, rteEvent.contextAtomicComponent, rte Event.targetRteEvent **InstanceRef implemented by:** RteEventInEcuInstance Ref
variableAccess	VariableAccess	*	iref	The VariableAccess for which the data buffer is used. **Tags:** atp.Splitkey=variableAccess, variable Access.contextRootComposition, variableAccess.context AtomicComponent, variableAccess.targetVariableAccess **InstanceRef implemented by:** VariableAccessInEcu InstanceRef

Table A.705: McDataAccessDetails

Class	**McDataInstance**			
Note	Describes the specific properties of one data instance in order to support measurement and/or calibration of this data instance. The most important attributes are: • Its shortName is copied from the ECU Flat map (if applicable) and will be used as identifier and for display by the MC system. • The category is copied from the corresponding data type (ApplicationDataType if defined, otherwise ImplementationDataType) as far as applicable. • The symbol is the one used in the programming language. It will be used to find out the actual memory address by the final generation tool with the help of linker generated information. It is assumed that in the M1 model this part and all the aggregated and referred elements (with the exception of the Flat Map and the references from ImplementationElementInParameterInstanceRef and McAccessDetails) are completely generated from "upstream" information. This means, that even if an element like e.g. a CompuMethod is only used via reference here, it will be copied into the M1 artifact which holds the complete McSupportData for a given Implementation.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	McDataInstance.subElement, McSupportData.mcParameterInstance, McSupportData.mcVariableInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arraySize	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of number of elements.
displayIdentifier	McdIdentifier	0..1	attr	An optional attribute to be used to set the ASAM ASAP2 DISPLAY_IDENTIFIER attribute.
flatMapEntry	FlatInstanceDescriptor	0..1	ref	Reference to the corresponding entry in the ECU Flat Map. This allows to trace back to the original specification of the generated data instance. This link shall be added by the RTE generator mainly for documentation purposes. The reference is optional because • The McDataInstance may represent an array or struct in which only the subElements correspond to FlatMap entries. • The McDataInstance may represent a task local buffer for rapid prototyping access which is different from the "main instance" used for measurement access. **Tags:** atp.Splitkey=flatMapEntry This Attribute is only used by the AUTOSAR Classic Platform.
instanceIn Memory	ImplementationElement InParameterInstance Ref	0..1	aggr	Reference to the corresponding data instance in the description of calibration data structures published by the RTE generator. This is used to support emulation methods inside the ECU, it is not required for A2L generation. **Tags:** atp.Splitkey=instanceInMemory, instanceIn Memory.context, instanceInMemory.target
mcDataAccess Details	McDataAccessDetails	0..1	aggr	Refers to "upstream" information on how the RTE uses this data instance. Use Case: Rapid Prototyping **Tags:** atp.Splitkey=mcDataAccessDetails
mcData Assignment	RoleBasedMcData Assignment	*	aggr	An assignment between McDataInstances. This supports the indication of related McDataElement implementing the of "RP global buffer", "RP global measurement buffer", "RP enabler flag". **Tags:** atp.Splitkey=mcDataAssignment

▽

△

Class	**McDataInstance**			
resulting Properties	SwDataDefProps	0..1	aggr	These are the generated properties resulting from decisions taken by the RTE generator for the actually implemented data instance. Only those properties are relevant here, which are needed for the measurement and calibration system. **Tags:** atp.Splitkey=resultingProperties
resultingRptSw Prototyping Access	RptSwPrototyping Access	0..1	aggr	Describes the implemented accessibility of data and modes by the rapid prototyping tooling. **Tags:** atp.Splitkey=resultingRptSwPrototypingAccess atp.Status=obsolete
role	Identifier	0..1	attr	An optional attribute to be used for additional information on the role of this data instance, for example in the context of rapid prototyping.
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the implemented code preparation for rapid prototyping at data accesses for a hook based bypassing. **Tags:** atp.Splitkey=rptImplPolicy atp.Status=obsolete
subElement (ordered)	McDataInstance	*	aggr	This relation indicates, that the target element is part of a "struct" which is given by the source element. This information will be used by the final generator to set up the correct addressing scheme. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
symbol	SymbolString	0..1	attr	This String is used to determine the memory address during final generation of the MC configuration data (e.g. "A2L" file) . It shall be the name of the element in the programming language such that it can be identified in linker generated information. In case the McDataInstance is part of composite data in the programming language, the symbol String may include parts denoting the element context, unless the context is given by the symbol attribute of an enclosing McDataInstance. This means in particular for the C language that the "." character shall be used as a separator between the name of a "struct" variable the name of one of its elements. The symbol can differ from the shortName in case of generated C data declarations. It is an optional attribute since it may be missing in case the instance represents an element (e.g. a single array element) which has no name in the linker map.

Table A.706: McDataInstance

Class	**McFunction**
Note	Represents a functional element to be used as input to support measurement and calibration. It is used to • assign calibration parameters to a logical function • assign measurement variables to a logical function • structure functions hierarchically **Tags:** atp.recommendedPackage=McFunctions
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*

▽

△

Class	**McFunction**			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
defCalprmSet	McFunctionDataRefSet	0..1	aggr	Refers to the set of adjustable data (= calibration parameters) defined in this function. **Tags:** atp.Splitkey=defCalprmSet xml.sequenceOffset=10
inMeasurement Set	McFunctionDataRefSet	0..1	aggr	Refers to the set of measurable input data for this function. **Tags:** atp.Splitkey=inMeasurementSet xml.sequenceOffset=30
loc Measurement Set	McFunctionDataRefSet	0..1	aggr	Refers to the set of measurable local data in this function. **Tags:** atp.Splitkey=locMeasurementSet xml.sequenceOffset=50
out Measurement Set	McFunctionDataRefSet	0..1	aggr	Refers to the set of measurable output data from this function. **Tags:** atp.Splitkey=outMeasurementSet xml.sequenceOffset=60
refCalprmSet	McFunctionDataRefSet	0..1	aggr	Refers to the set of adjustable data (= calibration parameters) referred by this function. **Tags:** atp.Splitkey=refCalprmSet xml.sequenceOffset=20
subFunction	McFunction	*	ref	A sub-function that is seen as part of the enclosing function. **Tags:** atp.Splitkey=subFunction xml.sequenceOffset=70

Table A.707: McFunction

Class	«atpVariation» **McFunctionDataRefSet**			
Note	Refers to a set of data assigned to an McFunction in a particular role. The data are given • either by entries in a FlatMap • or by data instances that are part of MC support data. These two possibilities are exclusive within a given McFunctionDataRefSet. Which one to use depends on the process and tool environment. The set is subject to variability because the same functional model may be used with various representation of the data. **Tags:** vh.latestBindingTime=preCompileTime			
Base	*ARObject*			
Aggregated by	McFunction.defCalprmSet, McFunction.inMeasurementSet, McFunction.locMeasurementSet, McFunction.outMeasurementSet, McFunction.refCalprmSet			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	«atpVariation» **McFunctionDataRefSet**			
flatMapEntry	FlatInstanceDescriptor	*	ref	Refers to an entry in a FlatMap that is part of the set, for example a calibration parameter or measured variable. **Tags:** atp.Splitkey=flatMapEntry xml.sequenceOffset=10 This Attribute is only used by the AUTOSAR Classic Platform.
mcDataInstance	McDataInstance	*	ref	Refers to a data instance within MC support data that is part of the set, i.e. a calibration parameter or measured variable. **Tags:** atp.Splitkey=mcDataInstance xml.sequenceOffset=20

Table A.708: McFunctionDataRefSet

Class	**McGroup**			
Note	Represents a group element to be used as input to support measurement and calibration. It is used to provide selection lists (groups) of calibration parameters, measurement variables, and functions in a hierarchical manner (subGroups). **Tags:** atp.recommendedPackage=McFunctions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mcFunction	McFunction	*	ref	A McFunction that is seen as part of the enclosing group. **Tags:** atp.Splitkey=mcFunction xml.sequenceOffset=40
refCalprmSet	McGroupDataRefSet	0..1	aggr	Refers to the set of adjustable data (= calibration parameters) referred by this McGroup. **Tags:** atp.Splitkey=refCalprmSet xml.sequenceOffset=20
ref Measurement Set	McGroupDataRefSet	0..1	aggr	Refers to the set of measurable belonging to this Mc Group. **Tags:** atp.Splitkey=refMeasurementSet xml.sequenceOffset=30
subGroup	McGroup	*	ref	A sub-group that is seen as part of the enclosing group. **Tags:** atp.Splitkey=subGroup xml.sequenceOffset=10

Table A.709: McGroup

Class	«atpVariation» **McGroupDataRefSet**			
Note	Refers to a set of data assigned to an McGroup in a particular role. The data are given • either by entries in a FlatMap • or by data instances that are part of MC support data. These two possibilities can be mixed within a given McGroupDataRefSet. Which one to use depends on the process and tool environment. The set is subject to variability because the same functional model may be used with various representation of the data. **Tags:** vh.latestBindingTime=preCompileTime			
Base	*ARObject*			
Aggregated by	McGroup.refCalprmSet, McGroup.refMeasurementSet			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
flatMapEntry	FlatInstanceDescriptor	*	ref	Refers to an entry in a FlatMap that is part of the set, for example a calibration parameter or measured variable. **Tags:** atp.Splitkey=flatMapEntry xml.sequenceOffset=50 This Attribute is only used by the AUTOSAR Classic Platform.
mcDataInstance	McDataInstance	*	ref	Refers to a data instance within MC support data that is part of the set, i.e. a calibration parameter or measured variable. **Tags:** atp.Splitkey=mcDataInstance xml.sequenceOffset=60

Table A.710: McGroupDataRefSet

Class	**McParameterElementGroup**			
Note	Denotes a group of calibration parameters which are handled by the RTE as one data structure.			
Base	*ARObject*			
Aggregated by	McSwEmulationMethodSupport.elementGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ramLocation	VariableDataPrototype	0..1	ref	Refers to the RAM location of this parameter group. To be used for the init-RAM method. **Tags:** atp.Splitkey=ramLocation
romLocation	ParameterData Prototype	0..1	ref	Refers to the ROM location of this parameter group. To be used for the init-RAM method. **Tags:** atp.Splitkey=romLocation
shortLabel	Identifier	0..1	attr	Assigns a name to this element. **Tags:** xml.sequenceOffset=-100

Table A.711: McParameterElementGroup

Class	**McSupportData**			
Note	Root element for all measurement and calibration support data related to one Implementation artifact on an ECU. There shall be one such element related to the RTE implementation (if it owns MC data) and a separate one for each module or component, which owns private MC data.			
Base	*ARObject*			
Aggregated by	*Implementation*.mcSupport			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**McSupportData**			
emulation Support	McSwEmulationMethod Support	*	aggr	Describes the calibration method used by the RTE. This information is not needed for A2L generation, but to setup software emulation in the ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=emulationSupport, emulation Support.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
mcParameter Instance	McDataInstance	*	aggr	A data instance to be used for calibration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mcParameterInstance, mcParameter Instance.shortName, mcParameterInstance.variation Point.shortLabel vh.latestBindingTime=postBuild
mcVariable Instance	McDataInstance	*	aggr	A data instance to be used for measurement. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mcVariableInstance, mcVariable Instance.shortName, mcVariableInstance.variation Point.shortLabel vh.latestBindingTime=postBuild
measurable System ConstantValues	SwSystemconstant ValueSet	*	ref	Sets of system constant values to be transferred to the MCD system, because the system constants have been specified with "swCalibrationAccess" = readonly. **Tags:** atp.Splitkey=measurableSystemConstantValues
rptSupportData	RptSupportData	0..1	aggr	The rapid prototyping support data belonging to this implementation. The aggregtion is <<atpSplitable>> because in case of an already exisiting BSW Implementation model, this description will be added later in the process, namely at code generation time. **Tags:** atp.Splitkey=rptSupportData

Table A.712: McSupportData

Class	**McSwEmulationMethodSupport**			
Note	This denotes the method used by the RTE to handle the calibration data. It is published by the RTE generator and can be used e.g. to generate the corresponding emulation method in a Complex Driver. According to the actual method given by the category attribute, not all attributes are always needed: • double pointered method: only baseReference is mandatory • single pointered method: only referenceTable is mandatory • initRam method: only elementGroup(s) are mandatory Note: For single/double pointered method the group locations are implicitly accessed via the reference table and their location can be found from the initial values in the M1 model of the respective pointers. Therefore, the description of elementGroups is not needed in these cases. Likewise, for double pointered method the reference table description can be accessed via the M1 model under baseReference.			
Base	*ARObject*			
Aggregated by	McSupportData.emulationSupport			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseReference	VariableDataPrototype	0..1	ref	Refers to the base pointer in case of the double-pointered method. **Tags:** atp.Splitkey=baseReference

▽

△

Class	**McSwEmulationMethodSupport**			
category	Identifier	0..1	attr	Identifies the actual method. The possible names shall correspond to the symbols of the ECU configuration parameter for the calibration method of the RTE, and can include vendor specific methods. **Tags:** xml.sequenceOffset=-90
elementGroup	McParameterElement Group	*	aggr	Denotes the grouping of calibration parameters in the actual RTE code. Depending on the category, this information maybe required to set up the emulation code. **Tags:** atp.Splitkey=elementGroup
referenceTable	VariableDataPrototype	0..1	ref	Refers to the pointer table in case of the single-pointered method. **Tags:** atp.Splitkey=referenceTable
shortLabel	Identifier	0..1	attr	Assigns a name to this element. **Tags:** xml.sequenceOffset=-100

Table A.713: McSwEmulationMethodSupport

Class	**MeasuredExecutionTime**			
Note	Specifies the ExecutionTime which has been gathered using measurement means.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
maximum ExecutionTime	MultidimensionalTime	0..1	aggr	The maximum measured execution time. **Tags:** atp.Splitkey=maximumExecutionTime
minimum ExecutionTime	MultidimensionalTime	0..1	aggr	The minimum measured execution time. **Tags:** atp.Splitkey=minimumExecutionTime
nominal ExecutionTime	MultidimensionalTime	0..1	aggr	The nominal measured execution time. **Tags:** atp.Splitkey=nominalExecutionTime

Table A.714: MeasuredExecutionTime

Class	**MeasuredHeapUsage**			
Note	The heap usage has been measured.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
averageMemory Consumption	PositiveInteger	0..1	attr	The average heap usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	0..1	attr	The maximum heap usage measured. Unit: byte.
minimum Memory Consumption	PositiveInteger	0..1	attr	The minimum heap usage measured. Unit: byte.
testPattern	String	0..1	attr	Description of the test pattern used to acquire the measured values.

Table A.715: MeasuredHeapUsage

Class	**MeasuredStackUsage**			
Note	The stack usage has been measured.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
averageMemory Consumption	PositiveInteger	0..1	attr	The average stack usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	0..1	attr	The maximum stack usage measured. Unit: byte.
minimum Memory Consumption	PositiveInteger	0..1	attr	The minimum stack usage measured. Unit: byte.
testPattern	String	0..1	attr	Description of the test pattern used to acquire the measured values.

Table A.716: MeasuredStackUsage

Enumeration	**MemoryAllocationKeywordPolicyType**
Note	Enumeration to specify the name pattern of the Memory Allocation Keyword.
Aggregated by	SwAddrMethod.memoryAllocationKeywordPolicy
Literal	***Description***
addrMethodShort Name	The MemorySection shortNames of referring MemorySections and therefore the belonging Memory Allocation Keywords in the code are build with the shortName of the SwAddrMethod. This is the default value if the attribute does not exist. **Tags:** atp.EnumerationLiteralIndex=0
addrMethodShort NameAndAlignment	The MemorySection shortNames of referring MemorySections and therefore the belonging Memory Allocation Keywords in the code are build with the shortName of the SwAddrMethod and a variable alignment postfix. Thereby the alignment postfix needs to be consistent with the alignment attribute of the related MemorySection. **Tags:** atp.EnumerationLiteralIndex=1

Table A.717: MemoryAllocationKeywordPolicyType

Class	**MemorySection**
Note	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. 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: <SwAddrMethod shortName>[_<further specialization nominator>][_<alignment>] where • **[<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 memory AllocationKeywordPolicy value of the referenced SwAddrMethod is set to addrMethodShortNameAnd Alignment

△

Class	MemorySection			
	△ 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. 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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.memorySection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
alignment	AlignmentType	0..1	attr	The attribute describes the typical alignment of objects within this memory section.
executableEntity	ExecutableEntity	*	ref	Reference to the ExecutableEntitites located in this section. This allows to locate different Executable Entitities in different sections even if the associated Sw Addrmethod is the same. This is applicable to code sections only. **Tags:** atp.Splitkey=executableEntity
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. **Tags:** atp.Splitkey=prefix
size	PositiveInteger	0..1	attr	The size in bytes of the section.
swAddrmethod	SwAddrMethod	0..1	ref	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. 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. **Tags:** atp.Splitkey=swAddrmethod
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 A.718: MemorySection

Class	MemorySectionLocation			
Note	Specifies in which hardware ProvidedMemorySegment the softwareMemorySection is located.			
Base	*ARObject*			
Aggregated by	*ExecutionTime*.memorySectionLocation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provided Memory	HwElement	0..1	ref	Reference to the hardware ProvidedMemorySegment. **Tags:** atp.Splitkey=providedMemory
software MemorySection	MemorySection	0..1	ref	Reference to the MemorySection which is mapped on a certain hardware memory segment. **Tags:** atp.Splitkey=softwareMemorySection

Table A.719: MemorySectionLocation

Class	MessageToBypassCanSecCommunicationMapping			
Note	Mapping of FrameTriggerings and SoConIPduIdentifiers to explicitly be bypassed communication. **Tags:** atp.recommendedPackage=SecureChannelMappings			
Base	*ARElement*, *ARObject*, *AbstractMessageToSecureChannelMapping*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
can Communication Connector	CanCommunication Connector	*	ref	Reference to the CanCommunicationConnectors on which the messages shall be bypassed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canCommunicationConnector, can CommunicationConnector.canCommunicationConnector, canCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.720: MessageToBypassCanSecCommunicationMapping

Class	MessageToCanSecSecureChannelMapping			
Note	Mapping of FrameTriggerings and SoConIPduIdentifiers to a CanSecSecureChannel. **Tags:** atp.recommendedPackage=SecureChannelMappings			
Base	*ARElement*, *ARObject*, *AbstractMessageToSecureChannelMapping*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
secureChannel	CanSecSecureChannel	0..1	ref	Reference to the SecureChannel to which the Frame Triggerings are mapped. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secureChannel, secureChannel.canSec SecureChannel, secureChannel.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.721: MessageToCanSecSecureChannelMapping

Class	**MetaDataItem**			
Note	This meta-class represents a single meta-data item.			
Base	*ARObject*			
Aggregated by	MetaDataItemSet.metaDataItem			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
length	PositiveInteger	0..1	attr	This attribute determines the length of the `MetaDataItem` at run-time. The length is given in bytes.
metaDataItem Type	TextValueSpecification	0..1	aggr	This aggregation contributes the specification of the concrete meta-data item type.

Table A.722: MetaDataItem

Class	**MetaDataItemSet**			
Note	This meta-class represents the ability to define a set of meta-data items to be used in `SenderReceiverInterface`s.			
Base	*ARObject*			
Aggregated by	SenderReceiverInterface.metaDataItemSet			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	ref	This reference identifies the dataElement for which the ordered list of meta-data items is defined. **Tags:** atp.Splitkey=dataElement
metaDataItem (ordered)	MetaDataItem	*	aggr	This aggregation represents the ordered definition of meta-data items.

Table A.723: MetaDataItemSet

Enumeration	**MirroringProtocolEnum**
Note	Eunumeration that defines the supported bus mirroring protocol options) with two literals. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*BusMirrorChannelMapping*.mirroringProtocol
Literal	***Description***
none	mirroringProtocol is not used **Tags:** atp.EnumerationLiteralIndex=1
version1	version1 of the mirroringProtocol is used **Tags:** atp.EnumerationLiteralIndex=0

Table A.724: MirroringProtocolEnum

Class	«atpMixedString» ***MixedContentForParagraph*** (abstract)			
Note	This mainly represents the text model of a full blown paragraph within a documentation.			
Base	*ARObject*			
Subclasses	LParagraph, SlParagraph			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
br	Br	0..1	aggr	This element is the same as function here as in a HTML document i.e. it forces a line break. **Tags:** xml.sequenceOffset=40
e	EmphasisText	0..1	aggr	This is emphasized text. **Tags:** xml.sequenceOffset=70

▽

△

Class	«atpMixedString» ***MixedContentForParagraph*** (abstract)			
ft	SlParagraph	0..1	aggr	This is a foot note within a paragraph.
ie	IndexEntry	0..1	aggr	This is an index entry. **Tags:** xml.sequenceOffset=110
std	Std	0..1	aggr	This is a refeernce to a standard. **Tags:** xml.sequenceOffset=120
sub	Superscript	0..1	attr	This is subscript text. **Tags:** xml.sequenceOffset=100
sup	Superscript	0..1	attr	This is superscript text. **Tags:** xml.sequenceOffset=90
trace	Traceable	0..1	ref	This allows to place an arbitrary reference to a traceable object in documentation.
tt	Tt	0..1	aggr	This is a technical term. **Tags:** xml.sequenceOffset=30
xdoc	Xdoc	0..1	aggr	This is a reference to a printable external document. **Tags:** xml.sequenceOffset=130
xfile	Xfile	0..1	aggr	This represents a reference to an external file which usually cannot be printed. **Tags:** xml.sequenceOffset=140
xref	Xref	0..1	aggr	This is a cross reference. **Tags:** xml.sequenceOffset=50
xrefTarget	XrefTarget	0..1	aggr	This element specifies a reference target which can be scattered throughout the text. **Tags:** xml.sequenceOffset=60

Table A.725: MixedContentForParagraph

Class	**ModeAccessPoint**			
Note	A ModeAccessPoint is required by a RunnableEntity owned by a Mode Manager or Mode User. Its semantics implies the ability to access the current mode (provided by the RTE) of a ModeDeclaration GroupPrototype's ModeDeclarationGroup.			
Base	*ARObject*			
Aggregated by	RunnableEntity.modeAccessPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ident	ModeAccessPointIdent	0..1	aggr	The aggregation in the role ident provides the ability to make the ModeAccessPoint identifiable. From the semantical point of view, the ModeAccessPoint is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=-100

▽

△

Class	**ModeAccessPoint**			
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is accessed by this runnable. **Tags:** atp.Splitkey=modeGroup, modeGroup.contextPPort, modeGroup.contextRPort, modeGroup.targetModeGroup xml.typeElement=true **InstanceRef implemented by:** ModeGroupInAtomicSwc InstanceRef

Table A.726: ModeAccessPoint

Enumeration	**ModeActivationKind**
Note	Kind of mode switch condition used for activation of an event, as further described for each enumeration field.
Aggregated by	BswModeSwitchEvent.activation, SwcModeSwitchEvent.activation
Literal	*Description*
onEntry	On entering the referred mode. **Tags:** atp.EnumerationLiteralIndex=0
onExit	On exiting the referred mode. **Tags:** atp.EnumerationLiteralIndex=1
onTransition	On transition of the 1st referred mode to the 2nd referred mode. **Tags:** atp.EnumerationLiteralIndex=2

Table A.727: ModeActivationKind

Class	**ModeDeclaration**			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationGroup.modeDeclaration			
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 A.728: ModeDeclaration

Class	**ModeDeclarationGroup**			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.recommendedPackage=ModeDeclarationGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
initialMode	ModeDeclaration	0..1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred. **Tags:** atp.Splitkey=initialMode

▽

△

Class	**ModeDeclarationGroup**			
mode Declaration	ModeDeclaration	*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeDeclaration, modeDeclaration.shortName, modeDeclaration.variationPoint.shortLabel 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). **Tags:** atp.Splitkey=modeManagerErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.
modeTransition	ModeTransition	*	aggr	This represents the avaliable ModeTransitions of the ModeDeclarationGroup **Tags:** atp.Splitkey=modeTransition, modeTransition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
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). **Tags:** atp.Splitkey=modeUserErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.
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. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.729: ModeDeclarationGroup

Class	**ModeDeclarationGroupPrototype**			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.providedModeGroup, BswModuleDescription.requiredModeGroup, FirewallStateSwitchInterface.firewallStateMachine, FunctionGroupSet.functionGroup, ModeSwitchInterface.modeGroup, Process.processStateMachine, StateManagementStateNotification.stateMachine			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swCalibration Access	SwCalibrationAccessEnum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time. This Attribute is only used by the AUTOSAR Classic Platform.
type	ModeDeclarationGroup	0..1	tref	The "collection of ModeDeclarations" (= ModeDeclaration Group) supported by a component **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table A.730: ModeDeclarationGroupPrototype

Class	ModeDeclarationGroupPrototypeMapping			
Note	Defines the mapping of two particular ModeDeclarationGroupPrototypes (in the given context) that are unequally named and/or require a reference to a ModeDeclarationMappingSet in order to become compatible by definition of ModeDeclarationMappings.			
Base	*ARObject*			
Aggregated by	ModeInterfaceMapping.modeMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstModeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype to be mapped. **Tags:** atp.Splitkey=firstModeGroup
mode Declaration MappingSet	ModeDeclaration MappingSet	0..1	ref	This represents the available mappings of Mode Declarations in the context ot this ModeDeclarationGroup Prototype. **Tags:** atp.Splitkey=modeDeclarationMappingSet
secondMode Group	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype to be mapped. **Tags:** atp.Splitkey=secondModeGroup

Table A.731: ModeDeclarationGroupPrototypeMapping

Class	ModeDeclarationMapping			
Note	This meta-class implements a concrete mapping of two `ModeDeclaration`s.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationMappingSet.modeDeclarationMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstMode	ModeDeclaration	*	ref	This represents the first ModeDeclaration of the Mode DeclarationMapping. This reference has the multiplicity 0 ..* to support use cases where e.g. one mode of the mode user is mapped to several modes of the mode manager. **Tags:** atp.Splitkey=firstMode
secondMode	ModeDeclaration	0..1	ref	This represents the second ModeDeclaration of the Mode DeclarationMapping. **Tags:** atp.Splitkey=secondMode

Table A.732: ModeDeclarationMapping

Class	ModeDeclarationMappingSet			
Note	This meta-class implements a container for `ModeDeclarationMapping`s **Tags:** atp.recommendedPackage=PortInterfaceMappingSets			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode Declaration Mapping	ModeDeclaration Mapping	*	aggr	This represents the collection of ModeDeclaration Mappings owned by the enclosing ModeDeclaration MappingSet. **Tags:** atp.Splitkey=modeDeclarationMapping, mode DeclarationMapping.shortName

Table A.733: ModeDeclarationMappingSet

Class	**ModeDrivenTransmissionModeCondition**			
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*			
Aggregated by	TransmissionModeDeclaration.modeDrivenFalseCondition, TransmissionModeDeclaration.modeDriven TrueCondition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mode Declaration	ModeDeclaration	*	ref	Reference to one modeDeclaration which is OR associated in the context of the ModeDrivenTransmission ModeCondition. **Tags:** atp.Splitkey=modeDeclaration

Table A.734: ModeDrivenTransmissionModeCondition

Class	**ModeErrorBehavior**			
Note	This represents the ability to define the error behavior in the context of mode handling.			
Base	*ARObject*			
Aggregated by	ModeDeclarationGroup.modeManagerErrorBehavior, ModeDeclarationGroup.modeUserErrorBehavior			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultMode	ModeDeclaration	0..1	ref	This represents the ModeDeclaration that is considered the error mode in the context of the enclosing Mode DeclarationGroup. **Tags:** atp.Splitkey=defaultMode
errorReaction Policy	ModeErrorReaction PolicyEnum	0..1	attr	This represents the ability to define the policy in terms of which default model shall apply in case an error occurs.

Table A.735: ModeErrorBehavior

Enumeration	**ModeErrorReactionPolicyEnum**
Note	This represents the ability to specify the reaction on a mode error.
Aggregated by	ModeErrorBehavior.errorReactionPolicy
Literal	***Description***
defaultMode	This represents the ability to switch to the defaultMode in case of a mode error. **Tags:** atp.EnumerationLiteralIndex=0
lastMode	This represents the ability to keep the last mode in case of a mode error. **Tags:** atp.EnumerationLiteralIndex=1

Table A.736: ModeErrorReactionPolicyEnum

Class	**ModeInBswInstanceRef**			
Note	Instance reference to be capable of referencing a specific ModeDeclaration of a ModeDeclarationGroup Prototype utilized in a BSW module.			
Base	*ARObject*, *AtpInstanceRef*, *ModeInSwcBswInstanceRef*			
Aggregated by	TimingModeInstance.modeInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	ModeInBswInstanceRef			
contextBsw Implementation	BswImplementation	0..1	ref	Specifies the BSW implementation that manifests the context. **Tags:** xml.sequenceOffset=10 This Attribute is only used by the AUTOSAR Classic Platform.
contextMode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	ref	Specifies the mode declaration group prototype that manifests the context. **Tags:** xml.sequenceOffset=20
targetMode Declaration	ModeDeclaration	0..1	ref	Specifies the specific mode declaration in the given context. **Tags:** xml.sequenceOffset=30

Table A.737: ModeInBswInstanceRef

Class	ModeInSwcInstanceRef			
Note	Instance reference to be capable of referencing a ModeDeclaration at a specific Mode Switch Port of a SW-C.			
Base	*ARObject*, *AtpInstanceRef*, *ModeInSwcBswInstanceRef*			
Aggregated by	TimingModeInstance.modeInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	SwComponentType	0..1	ref	Specifies the SW component representing the base of the context. **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component	SwComponent Prototype	*	ref	Specifies the SW component prototype representing the context. **Tags:** xml.sequenceOffset=20
contextMode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	ref	Specifies the mode declaration group prototype that manifests the context. **Tags:** xml.sequenceOffset=40
contextPort	PortPrototype	0..1	ref	Specifies the port prototype representing the context. **Tags:** xml.sequenceOffset=30
targetMode Declaration	ModeDeclaration	0..1	ref	Specifies the specific mode declaration in the given context. **Tags:** xml.sequenceOffset=50

Table A.738: ModeInSwcInstanceRef

Class	ModeInterfaceMapping			
Note	Defines the mapping of ModeDeclarationGroupPrototypes in context of two different ModeInterfaces.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeMapping	ModeDeclarationGroup PrototypeMapping	0..1	aggr	Mapping of two ModeDeclarationGroupPrototypes in two different ModeInterfaces **Tags:** atp.Splitkey=modeMapping

Table A.739: ModeInterfaceMapping

Class	ModePortAnnotation			
Note	Annotation to a port used for calibration regarding a certain ModeDeclarationGroupPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.modePortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	The instance of annotated ModeDeclarationGroup Prototype. **Tags:** atp.Splitkey=modeGroup

Table A.740: ModePortAnnotation

Class	ModeRequestTypeMap			
Note	Specifies a mapping between a `ModeDeclarationGroup` and an `ImplementationDataType`. This `ImplementationDataType` shall be used to implement the `ModeDeclarationGroup`.			
Base	*ARObject*			
Aggregated by	DataTypeMappingSet.modeRequestTypeMap			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
implementation DataType	AbstractImplementation DataType	0..1	ref	This is the corresponding `AbstractImplementationDataType`. It shall be modeled along the idea of an "unsigned integer-like" data type. **Tags:** atp.Splitkey=implementationDataType
modeGroup	ModeDeclarationGroup	0..1	ref	This is the corresponding `ModeDeclarationGroup`. **Tags:** atp.Splitkey=modeGroup

Table A.741: ModeRequestTypeMap

Class	ModeSwitchEventTriggeredActivity			
Note	This meta-class defines an activity of the NvBlockSwComponentType for a specific NvBlock which is triggered by a ModeSwitchEvent.			
Base	*ARObject*			
Aggregated by	NvBlockDescriptor.modeSwitchEventTriggeredActivity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
role	Identifier	0..1	attr	This attribute indicates which service of the NvM for the NvBlock shall be requested.
swcModeSwitch Event	SwcModeSwitchEvent	0..1	ref	This reference identifies the SwcModeSwitchEvent that triggers the activity. **Tags:** atp.Splitkey=swcModeSwitchEvent

Table A.742: ModeSwitchEventTriggeredActivity

Class	ModeSwitchInterface			
Note	A mode switch interface declares a `ModeDeclarationGroupPrototype` to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	ModeSwitchInterface			
modeGroup	ModeDeclarationGroup Prototype	0..1	aggr	The ModeDeclarationGroupPrototype of this mode interface. **Tags:** atp.Splitkey=modeGroup, modeGroup.shortName

Table A.743: ModeSwitchInterface

Class	ModeSwitchPoint			
Note	A ModeSwitchPoint is required by a RunnableEntity owned a Mode Manager. Its semantics implies the ability to initiate a mode switch.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.modeSwitchPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is switched by this runnable. **Tags:** atp.Splitkey=modeGroup, modeGroup.context PPort, modeGroup.targetModeGroup **InstanceRef implemented by:** PModeGroupInAtomic SwcInstanceRef

Table A.744: ModeSwitchPoint

Class	ModeSwitchReceiverComSpec			
Note	Communication attributes of RPortPrototypes with respect to mode communication			
Base	*ARObject*, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to "true" the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype (of the same Port Interface) to which these communication attributes apply. **Stereotypes:** atpIdentityContributor
supports Asynchronous ModeSwitch	Boolean	0..1	attr	This attribute controls the behavior of the corresponding RPortPrototype with respect to the question whether it can deal with asynchronous mode switch requests, i.e. if set to true, the RPortPrototype is able to deal with an asynchronous mode switch request.

Table A.745: ModeSwitchReceiverComSpec

Class	ModeSwitchSenderComSpec			
Note	Communication attributes of PPortPrototypes with respect to mode communication			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**ModeSwitchSenderComSpec**			
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to "true" the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype (of the same Port Interface) to which these communication attributes apply. **Stereotypes:** atpIdentityContributor
modeSwitched Ack	ModeSwitchedAck Request	0..1	aggr	If this aggregation exists an acknowledgement for the successful processing of the mode switch request is required. **Tags:** atp.Splitkey=modeSwitchedAck
queueLength	PositiveInteger	0..1	attr	Length of call queue on the mode user side. The queue is implemented by the RTE. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.

Table A.746: ModeSwitchSenderComSpec

Class	**ModeSwitchedAckEvent**			
Note	This event is raised when the referenced ModeSwitchPoint has been processed or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventSource	ModeSwitchPoint	0..1	ref	The referenced ModeSwitchPoint raises this Mode SwitchedAckEvent when the ModeSwitchPoint has been processed. **Tags:** atp.Splitkey=eventSource

Table A.747: ModeSwitchedAckEvent

Class	**ModeSwitchedAckRequest**			
Note	Requests acknowledgements that a mode switch has been proceeded successfully			
Base	*ARObject*			
Aggregated by	ModeSwitchSenderComSpec.modeSwitchedAck			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.748: ModeSwitchedAckRequest

Class	**ModeTransition**			
Note	This meta-class represents the ability to describe possible ModeTransitions in the context of a Mode DeclarationGroup.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationGroup.modeTransition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ModeTransition**			
enteredMode	ModeDeclaration	0..1	ref	This represents the entered model of the ModeTransition. **Tags:** atp.Splitkey=enteredMode
exitedMode	ModeDeclaration	0..1	ref	This represents the exited mode of the ModeTransition **Tags:** atp.Splitkey=exitedMode

Table A.749: ModeTransition

Class	**MultidimensionalTime**			
Note	Specifies a time value based on [14] see [TPS_GST_00354].			
Base	*ARObject*			
Aggregated by	AgeConstraint.maximum, AgeConstraint.minimum, AnalyzedExecutionTime.bestCaseExecutionTime, AnalyzedExecutionTime.worstCaseExecutionTime, ArbitraryEventTriggering.maximumDistance, ArbitraryEventTriggering.minimumDistance, BurstPatternEventTriggering.minimumInterArrivalTime, BurstPatternEventTriggering.patternJitter, BurstPatternEventTriggering.patternLength, BurstPatternEventTriggering.patternPeriod, ConcretePatternEventTriggering.offset, ConcretePatternEventTriggering.patternJitter, ConcretePatternEventTriggering.patternLength, ConcretePatternEventTriggering.patternPeriod, ConfidenceInterval.lowerBound, ConfidenceInterval.upperBound, ExecutionTimeConstraint.maximum, ExecutionTimeConstraint.minimum, IoHwAbstractionServerAnnotation.age, LatencyTimingConstraint.maximum, LatencyTimingConstraint.minimum, LatencyTimingConstraint.nominal, MeasuredExecutionTime.maximumExecutionTime, MeasuredExecutionTime.minimumExecutionTime, MeasuredExecutionTime.nominalExecutionTime, OffsetTimingConstraint.maximum, OffsetTimingConstraint.minimum, PeriodicEventTriggering.jitter, PeriodicEventTriggering.minimumInterArrivalTime, PeriodicEventTriggering.period, ReceiverAnnotation.signalAge, RoughEstimateOfExecutionTime.estimatedExecutionTime, SimulatedExecutionTime.maximumExecutionTime, SimulatedExecutionTime.minimumExecutionTime, SimulatedExecutionTime.nominalExecutionTime, SporadicEventTriggering.jitter, SporadicEventTriggering.maximumInterArrivalTime, SporadicEventTriggering.minimumInterArrivalTime, SporadicEventTriggering.period, SwDataDefProps.swRefreshTiming, SynchronizationTimingConstraint.tolerance, TDLETZoneClock.accuracyExt, TDLETZoneClock.accuracyInt, TimingClockSyncAccuracy.accuracy, Trigger.triggerPeriod			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cseCode	CseCodeType	0..1	attr	Specifies the time base by means of CSE codes.
cseCodeFactor	Integer	0..1	attr	The scaling factor for the time value based on the specified CSE code.

Table A.750: MultidimensionalTime

Class	***MultilanguageReferrable*** (abstract)			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders). They also may have a longName. But they are not considered to contribute substantially to the overall structure of an AUTOSAR description. In particular it does not contain other Referrables.			
Base	*ARObject*, *Referrable*			
Subclasses	Caption, DefItem, DocumentationContext, *Identifiable*, SdgCaption, *TraceReferrable*, *Traceable*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
longName	MultilanguageLongName	0..1	aggr	This specifies the long name of the object. Long name is targeted to human readers and acts like a headline. **Tags:** atp.Splitkey=longName

Table A.751: MultilanguageReferrable

Class	**MultiplexedIPdu**			
Note	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. 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. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicPart	DynamicPart	0..1	aggr	According to the value of the selector field some parts of the IPdu have a different layout. In a complete System Description a MultiplexedIPdu shall contain a Dynamic Part. The following use cases support the multiplicity to be 0..1: • 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 MulitplexedIPdu 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. atpVariation: Content of a multiplexed PDU can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dynamicPart, dynamicPart.variation Point.shortLabel vh.latestBindingTime=postBuild
selectorField ByteOrder	ByteOrderEnum	0..1	attr	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. 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.
selectorField Length	Integer	0..1	attr	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 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.

▽

△

Class	**MultiplexedIPdu**			
selectorField StartPosition	Integer	0..1	attr	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.
staticPart	StaticPart	0..1	aggr	The static part of the multiplexed IPdu is the same regardless of the selector field. The static part is optional. atpVariation: Content of a multiplexed PDU can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticPart, staticPart.variationPoint.shortLabel vh.latestBindingTime=postBuild
triggerMode	TriggerMode	0..1	attr	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 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.
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPdu with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu. 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.

Table A.752: MultiplexedIPdu

Class	***MultiplexedPart*** (abstract)
Note	The StaticPart and the DynamicPart have common properties. Both can be separated in multiple segments within the multiplexed PDU.
Base	*ARObject*

▽

△

Class	**MultiplexedPart** (abstract)			
Subclasses	DynamicPart, StaticPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
segment Position	SegmentPosition	*	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. **Tags:** atp.Splitkey=segmentPosition

Table A.753: MultiplexedPart

Class	**NPdu**			
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	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.754: NPdu

Class	**NetworkEndpoint**			
Note	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetPhysicalChannel.networkEndpoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
fullyQualified DomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.
infrastructure Services	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed. **Tags:** atp.Splitkey=infrastructureServices This Attribute is only used by the AUTOSAR Classic Platform.
ipSecConfig	IPSecConfig	0..1	aggr	Optional IPSec configuration that provides security services for IP packets. **Tags:** atp.Splitkey=ipSecConfig
network Endpoint Address	NetworkEndpoint Address	*	aggr	Definition of a Network Address. **Tags:** atp.Splitkey=networkEndpointAddress, networkEndpoint Address.ipv4Address, networkEndpointAddress.ipv6 Address, networkEndpointAddress.macMulticastGroup xml.namePlural=NETWORK-ENDPOINT-ADDRESSES
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.

Table A.755: NetworkEndpoint

Class	**NetworkEndpointAddress** (abstract)			
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	*ARObject*			
Subclasses	Ipv4Configuration, Ipv6Configuration, MacMulticastConfiguration			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.756: NetworkEndpointAddress

Class	**NetworkSegmentIdentification**			
Note	This meta-class represents the ability to identify the PhysicalChannel on a system scope in a numerical way. One possible application of this approach is the Time Validation.			
Base	*ARObject*			
Aggregated by	GlobalTimeDomain.networkSegmentId			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
network SegmentId	PositiveInteger	0..1	attr	This attribute represents the numerical identifier of a PhysicalChannel on system level scope.

Table A.757: NetworkSegmentIdentification

Class	**NmCluster** (abstract)			
Note	Set of NM nodes coordinated with use of the NM algorithm.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmCluster, FlexrayNmCluster, J1939NmCluster, UdpNmCluster			
Aggregated by	NmConfig.nmCluster			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Cluster	CommunicationCluster	0..1	ref	Association to a CommunicationCluster in the topology description. **Tags:** atp.Splitkey=communicationCluster
nmChannel SleepMaster	Boolean	0..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. This Attribute is only used by the AUTOSAR Classic Platform.
nmLightTimeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left.
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmNode, nmNode.shortName, nmNode.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmNode Detection Enabled	Boolean	0..1	attr	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true. This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	***NmCluster*** (abstract)			
nmNodeId Enabled	Boolean	0..1	attr	Enables the source node identifier. This Attribute is only used by the AUTOSAR Classic Platform.
nmPnc Participation	Boolean	0..1	attr	Defines whether this NmCluster contributes to the partial network mechanism.
nmRepeatMsg IndEnabled	Boolean	0..1	attr	Switch for enabling the Repeat Message Bit Indication. This Attribute is only used by the AUTOSAR Classic Platform.
nm Synchronizing Network	Boolean	0..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. This Attribute is only used by the AUTOSAR Classic Platform.
pncCluster VectorLength	PositiveInteger	0..1	attr	Optionally defines the length of the PNC Request Bit Vector per CommunicationCluster (and VLAN in case of UdpNm). If not defined then System.pncVectorLength applies. Should only make the PNC Request Bit Vector shorter (or same length as defined in System.pncVectorLength).

Table A.758: NmCluster

Class	**NmConfig**			
Note	Contains the all configuration elements for AUTOSAR Nm. **Tags:** atp.recommendedPackage=NmConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmCluster	NmCluster	*	aggr	Collection of NM Clusters atpVariation: Derived, because cluster can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmCluster, nmCluster.shortName, nm Cluster.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmCluster Coupling	NmClusterCoupling	*	aggr	Collection of NmClusterCouplings atpVariation: Derived, because NmCluster can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmClusterCoupling, nmCluster Coupling.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	NmConfig			
nmIfEcu	NmEcu	*	aggr	Collection of NM ECUs atpVariation: Derived, because EcuInstance can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmIfEcu, nmIfEcu.shortName, nmIf Ecu.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

Table A.759: NmConfig

Class	NmEcu			
Note	ECU on which NM is running. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	NmConfig.nmIfEcu			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
busDependent NmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes **Tags:** atp.Splitkey=busDependentNmEcu
ecuInstance	EcuInstance	0..1	ref	Association to an ECUInstance in the topology description. **Tags:** atp.Splitkey=ecuInstance
nmBus Synchronization Enabled	Boolean	0..1	attr	Enables bus synchronization support.
nmComControl Enabled	Boolean	0..1	attr	Enables the Communication Control support.
nmCoordinator	NmCoordinator	0..1	aggr	Nm ECU may coordinate different clusters. **Tags:** atp.Splitkey=nmCoordinator
nmCycletime MainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the NM Interface in seconds.
nmPduRx Indication Enabled	Boolean	0..1	attr	Switch for enabling the PDU Rx Indication.
nmRemote SleepInd Enabled	Boolean	0..1	attr	Switch for enabling remote sleep indication support.
nmStateChange IndEnabled	Boolean	0..1	attr	Enables the CAN Network Management state change notification.
nmUserData Enabled	Boolean	0..1	attr	Switch for enabling user data support.

Table A.760: NmEcu

Class	***NmNode*** (abstract)
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Subclasses	CanNmNode, FlexrayNmNode, J1939NmNode, UdpNmNode

▽

△

Class	***NmNode*** (abstract)			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
controller	Communication Controller	0..1	ref	Association to an CommunicationController in the topology description. **Tags:** atp.Splitkey=controller This Attribute is only used by the AUTOSAR Classic Platform.
nmCoord Cluster	PositiveInteger	0..1	attr	NmCoordinationCluster identification number. This Attribute is only used by the AUTOSAR Classic Platform.
nmCoordinator Role	NmCoordinatorRole Enum	0..1	attr	This attribute indicates the role the NM Coordinator will have on this channel. This Attribute is only used by the AUTOSAR Classic Platform.
nmIfEcu	NmEcu	0..1	ref	Reference to the NmEcu that contains this NmNode. (CommunicationController that is referenced by the Nm Node shall be contained in the EcuInstance that is referenced by the NmEcu). **Tags:** atp.Splitkey=nmIfEcu This Attribute is only used by the AUTOSAR Classic Platform.
nmNodeId	Integer	0..1	attr	Node identifier of local NmNode. Shall be unique in the NmCluster.
nmPassive ModeEnabled	Boolean	0..1	attr	Enables support of the Passive Mode. The passive mode is configurable per channel. This Attribute is only used by the AUTOSAR Classic Platform.
nmVariant	NmVariantEnum	0..1	attr	Defines the functionality of Network Management. **Tags:** atp.Status=obsolete
rxNmPdu	NmPdu	*	ref	receive NM Pdu. **Tags:** atp.Splitkey=rxNmPdu This Attribute is only used by the AUTOSAR Classic Platform.
txNmPdu	NmPdu	*	ref	transmit NM Pdu **Tags:** atp.Splitkey=txNmPdu This Attribute is only used by the AUTOSAR Classic Platform.

Table A.761: NmNode

Class	**NmPdu**			
Note	Network Management Pdu **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**NmPdu**			
iSignalToIPdu Mapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUser Data 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. **Tags:** atp.Splitkey=iSignalToIPduMapping, iSignalToIPdu Mapping.shortName
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 nm DataInformation attribute is set to true.

Table A.762: NmPdu

Enumeration	**NmVariantEnum**
Note	Supported NmCoordinator roles.
Aggregated by	*CommunicationConnector*.nmVariant, *NmNode*.nmVariant
Literal	*Description*
full	AUTOSAR NM is available **Tags:** atp.EnumerationLiteralIndex=3
light	No AUTOSAR NM is available, but functionality to shut down a channel **Tags:** atp.EnumerationLiteralIndex=1
none	No NM available **Tags:** atp.EnumerationLiteralIndex=0
passive	AUTOSAR NM running in passive mode available **Tags:** atp.EnumerationLiteralIndex=2
slaveActive	No NM is available. This is used for e.g. LIN slaves **Tags:** atp.EnumerationLiteralIndex=4
slavePassive	No NM is available. This used for e.g. Ethernet communication channels with OA TC10 compliant hardware **Tags:** atp.EnumerationLiteralIndex=5

Table A.763: NmVariantEnum

Class	**NonqueuedReceiverComSpec**			
Note	Communication attributes specific to non-queued receiving.			
Base	*ARObject*, *RPortComSpec*, *ReceiverComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**NonqueuedReceiverComSpec**			
aliveTimeout	TimeValue	0..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. This Attribute is only used by the AUTOSAR Classic Platform.
enableUpdate	Boolean	0..1	attr	This attribute controls whether application code is entitled to check whether the value of the corresponding Variable DataPrototype has been updated. This Attribute is only used by the AUTOSAR Classic Platform.
filter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement. **Tags:** atp.Splitkey=filter
handleData Status	Boolean	0..1	attr	If this attribute is set to true, then 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. This Attribute is only used by the AUTOSAR Classic Platform.
handleNever Received	Boolean	0..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. • 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 Variable DataPrototype. This Attribute is only used by the AUTOSAR Classic Platform.
handleTimeout Type	HandleTimeoutEnum	0..1	attr	This attribute controls the behavior with respect to the handling of timeouts. This Attribute is only used by the AUTOSAR Classic Platform.
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, then the receiver's value will be used. **Tags:** atp.Splitkey=initValue This Attribute is only used by the AUTOSAR Classic Platform.
returnNoNew DataEnabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return No New Data Error if no new data is received from COM. This Attribute is only used by the AUTOSAR Classic Platform.
timeout Substitution Value	ValueSpecification	0..1	aggr	This attribute represents the substitution value applicable in the case of a timeout. **Tags:** atp.Splitkey=timeoutSubstitutionValue This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**NonqueuedReceiverComSpec**			
transportError CountEnabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return the number of transport errors (i.e. COM, SecOC errors) that happened since the last call of the respective API. This Attribute is only used by the AUTOSAR Classic Platform.
valueErrorCount Enabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return the number of value errors (i.e. out of range, invalid value) that happened since the last call of the respective API. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.764: NonqueuedReceiverComSpec

Class	**NonqueuedSenderComSpec**			
Note	Communication attributes for non-queued sender/receiver communication (sender side)			
Base	*ARObject*, *PPortComSpec*, *SenderComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement. **Tags:** atp.Splitkey=dataFilter
initValue	ValueSpecification	0..1	aggr	Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already. **Tags:** atp.Splitkey=initValue

Table A.765: NonqueuedSenderComSpec

Class	**NotAvailableValueSpecification**			
Note	This meta-class provides the ability to specify a ValueSpecification to state that the respective element is not available. This ability is needed to support the existence of ApplicationRecordElements where attribute isOptional ist set to the value true.			
Base	*ARObject*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultPattern	PositiveInteger	0..1	attr	The content of this attribute shall be used to initialize gaps in the memory occupied by a structured data type in the case that an NotAvailableValueSpecification is used. Note that this pattern is only applied during initialization!

Table A.766: NotAvailableValueSpecification

Class	**NumericalOrText**			
Note	This meta-class represents the ability to yield either a numerical or a string. A typical use case is that two or more instances of this meta-class are aggregated with a VariationPoint where some instances yield strings while other instances yield numerical depending on the resolution of the binding expression.			
Base	*ARObject*			
Aggregated by	RuleArguments.vtf, SwValues.vtf			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
vf	Numerical	0..1	attr	This attribute represents the ability to provide a numerical value. The latest binding time of the VariationPoint shall be preCompileTime. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
vt	String	0..1	attr	This attribute represents the ability to provide a textual value. **Tags:** xml.sequenceOffset=20

Table A.767: NumericalOrText

Class	**NumericalRuleBasedValueSpecification**			
Note	This meta-class is used to support a rule-based initialization approach for data types with an array-nature (ImplementationDataType of category ARRAY).			
Base	*ARObject*, *AbstractRuleBasedValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the array. **Tags:** atp.Splitkey=ruleBasedValues xml.roleElement=true xml.roleWrapperElement=false xml.typeWrapperElement=false

Table A.768: NumericalRuleBasedValueSpecification

Class	**NumericalValueSpecification**
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*

▽

△

Class	**NumericalValueSpecification**			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	Numerical	0..1	attr	This is the value itself. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.769: NumericalValueSpecification

Class	**NvBlockDataMapping**			
Note	Defines the mapping between the VariableDataPrototypes in the NvBlockComponents ports and the VariableDataPrototypes of the RAM Block. The data types of the referenced VariableDataPrototypes in the ports and the referenced sub-element (inside a CompositeDataType) of the VariableDataPrototype representing the RAM Block shall be compatible. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	BulkNvDataDescriptor.nvBlockDataMapping, NvBlockDescriptor.nvBlockDataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bitfieldTextTable MaskNvBlock Descriptor	PositiveInteger	0..1	attr	This attribute identifies the applicable bit mask on the side of the Nv Block.
bitfieldTextTable MaskPort Prototype	PositiveInteger	0..1	attr	This attribute identifies the applicable bit mask on the side of the PortPrototype.
nvRamBlock Element	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a RAM Block. **Tags:** atp.Splitkey=nvRamBlockElement
readNvData	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a pPort of the NvBlockComponent providing read access to the RAM Block.If there is no PortPrototype providing read access (write-only) the reference can be omitted. **Tags:** atp.Splitkey=readNvData
writtenNvData	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a rPort of the Nv BlockComponent providing write access to the RAM Block. If there is no port providing write access (read-only) the reference can be omitted. **Tags:** atp.Splitkey=writtenNvData
writtenReadNv Data	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a PRPort Prototype of the NvBlockSwComponentType providing write and read access to the RAM Block. **Tags:** atp.Splitkey=writtenReadNvData

Table A.770: NvBlockDataMapping

Class	**NvBlockDescriptor**			
Note	Specifies the properties of exactly one NVRAM Block. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, NvBlockSwComponentType.nvBlockDescriptor			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientServerPort	RoleBasedPort Assignment	*	aggr	The RoleBasedPortAssignement defines which client server port of the NvBlockSwComponentType serves for which kind of service or notification. In case of notifications one common callback function is provided by the RTE for each individual kind of notification defined by the "role". The aggregation of RoleBasedPortAssignment is subject to variability with the purpose to support the conditional existence of ports. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clientServerPort, clientServerPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular NVRAM Block **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular NVRAM Block. **Tags:** atp.Splitkey=dataTypeMapping
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	The purpose of InstantiationDataDefProps are the refinement of some data def properties of individual instantiations within the context of a NvBlockSw ComponentType. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of ports, component internal memory objects and those attributes. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeSwitch EventTriggered Activity	ModeSwitchEvent TriggeredActivity	*	aggr	This represents the collection of ModeSwitchEvent TriggeredActivities related to the enclosing NvBlock Descriptor. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeSwitchEventTriggeredActivity, mode SwitchEventTriggeredActivity.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**NvBlockDescriptor**			
nvBlockData Mapping	NvBlockDataMapping	*	aggr	Defines the mapping between the VariableData Prototypes in the NvBlockComponents ports and the VariableDataPrototypes of the RAM Block. The aggregation of NvBlockDataMapping is subject to variability with the purpose to support the conditional existence of nv data ports. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nvBlockDataMapping, nvBlockData Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
nvBlockNeeds	NvBlockNeeds	0..1	aggr	Specifies the abstract needs on the configuration of the NVRAM Manager for the single NVRAM Block described by this NvBlockDescriptor. In addition, it may define requirements for writing strategies in an implementation of an NvBlockSw ComponentType by the RTE. Please note that the attributes nDataSets and nRom Blocks are not relevant for this aggregation because the RTE will allocate just one block anyway. In a different context, however, they do make sense. **Tags:** atp.Splitkey=nvBlockNeeds, nvBlockNeeds.short Name
ramBlock	VariableDataPrototype	0..1	aggr	Defines the RAM Block of the NVRAM Block provided by NvBlockSwComponentType. **Tags:** atp.Splitkey=ramBlock, ramBlock.shortName
romBlock	ParameterData Prototype	0..1	aggr	Defines the ROM Block of the NVRAM Block provided by NvBlockSwComponentType. **Tags:** atp.Splitkey=romBlock, romBlock.shortName
supportDirty Flag	Boolean	0..1	attr	Specifies whether calling of NvM functions for writing and/ or status control of potentially modified RAM Blocks to NV memory shall be controlled by the RTE.
timingEvent	TimingEvent	0..1	ref	this reference can be taken to identify the TimingEvent to be used by the RTE for implementing a cyclic writing strategy for this block **Tags:** atp.Splitkey=timingEvent
writingStrategy	RoleBasedData Assignment	*	aggr	This attribute allows for assigning a specific writing strategy for an incoming AutosarDataPrototype. **Tags:** atp.Splitkey=writingStrategy

Table A.771: NvBlockDescriptor

Class	**NvBlockNeeds**			
Note	Specifies the abstract needs on the configuration of a single NVRAM Block.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, NvBlockDescriptor.nvBlockNeeds, SwcService Dependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calcRamBlock Crc	Boolean	0..1	attr	Defines if CRC (re)calculation for the permanent RAM Block is required.
checkStatic BlockId	Boolean	0..1	attr	Defines if the Static Block Id check shall be enabled.

▽

△

Class	**NvBlockNeeds**			
cyclicWriting Period	TimeValue	0..1	attr	This represents the period for cyclic writing of NvData to store the associated RAM Block.
nDataSets	PositiveInteger	0..1	attr	Number of data sets to be provided by the NVRAM manager for this block. This is the total number of ROM Blocks and RAM Blocks.
nRomBlocks	PositiveInteger	0..1	attr	Number of ROM Blocks to be provided by the NVRAM manager for this block. Please note that these multiple ROM Blocks are given in a contiguous area.
ramBlockStatus Control	RamBlockStatusControl Enum	0..1	attr	This attribute defines how the management of the RAM Block status is controlled.
readonly	Boolean	0..1	attr	true: data of this NVRAM Block are write protected for normal operation (but protection can be disabled) false: no restriction
reliability	NvBlockNeeds ReliabilityEnum	0..1	attr	Reliability against data loss on the non-volatile medium.
resistantTo ChangedSw	Boolean	0..1	attr	Defines whether an NVRAM Block shall be treated resistant to configuration changes (true) or not (false). For details how to handle initialization in the latter case, please refer to the NVRAM specification.
restoreAtStart	Boolean	0..1	attr	Defines whether the associated RAM Block shall be implicitly restored during startup by the basic software.
selectBlockFor FirstInitAll	Boolean	0..1	attr	If this attribute is set to true the NvM shall process this block in the NvM_FirstInitAll() function.
storeAt Shutdown	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored during shutdown by the basic software.
storeCyclic	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored periodically by the basic software.
store Emergency	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored in case of ECU failure (e.g. loss of power) by the basic software. If the attribute store Emergency is set to true the associated RAM Block shall be configured to have immediate priority.
storeImmediate	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored immediately during or after execution of the according SW-C RunnableEntity by the basic software.
storeOnChange	Boolean	0..1	attr	This attribute defines whether the associated RAM Block shall be stored immediately if the written value is different to the value stored in the associated RAM Block(s) during or after execution of the according SW-C RunnableEntity.
useAuto ValidationAt ShutDown	Boolean	0..1	attr	If set to true the RAM Block shall be auto validated during shutdown phase.
useCRCComp Mechanism	Boolean	0..1	attr	If set to true the CRC of the RAM Block shall be compared during a write job with the CRC which was calculated during the last successful read or write job in order to skip unnecessary NVRAM writings.
writeOnlyOnce	Boolean	0..1	attr	Defines write protection after first write: true: This block is prevented from being changed/erased or being replaced with the default ROM data after first initialization by the software-component. false: No such restriction.
writeVerification	Boolean	0..1	attr	Defines if Write Verification shall be enabled for this NVRAM Block.

▽

△

Class	**NvBlockNeeds**			
writing Frequency	PositiveInteger	0..1	attr	Provides the amount of updates to this block from the application point of view. It has to be provided in "number of write access per year".
writingPriority	NvBlockNeedsWriting PriorityEnum	0..1	attr	Requires the priority of writing this block in case of concurrent requests to write other blocks.

Table A.772: NvBlockNeeds

Enumeration	**NvBlockNeedsReliabilityEnum**
Note	Reliability against data loss on the non-volatile medium. These requirements give only a relative indication, for example on the required degree of redundancy for storage. They do, however, not specify by which means (e.g. software or hardware) the reliability is actually achieved.
Aggregated by	NvBlockNeeds.reliability
Literal	***Description***
errorCorrection	Errors shall be corrected **Tags:** atp.EnumerationLiteralIndex=0
errorDetection	Errors shall be detected **Tags:** atp.EnumerationLiteralIndex=1
noProtection	Data need not to be handled with protection **Tags:** atp.EnumerationLiteralIndex=2

Table A.773: NvBlockNeedsReliabilityEnum

Class	**NvBlockSwComponentType**			
Note	The NvBlockSwComponentType defines non volatile data which data can be shared between Sw ComponentPrototypes. The non volatile data of the NvBlockSwComponentType are accessible via provided and required ports. **Tags:** atp.recommendedPackage=SwComponentTypes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bulkNvData Descriptor	BulkNvDataDescriptor	*	aggr	This aggregation formally defines the bulk Nv Blocks that are provided to the application software by the enclosing NvBlockSwComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bulkNvDataDescriptor, bulkNvData Descriptor.shortName, bulkNvDataDescriptor.variation Point.shortLabel vh.latestBindingTime=preCompileTime
nvBlock Descriptor	NvBlockDescriptor	*	aggr	Specification of the properties of exactly one NVRAM Block. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nvBlockDescriptor, nvBlockDescriptor.short Name, nvBlockDescriptor.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.774: NvBlockSwComponentType

Class	**NvDataInterface**			
Note	A non volatile data interface declares a number of VariableDataPrototypes to be exchanged between non volatile block components and atomic software components. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nvData	VariableDataPrototype	*	aggr	The VariableDataPrototype of this nv data interface. **Tags:** atp.Splitkey=nvData, nvData.shortName

Table A.775: NvDataInterface

Class	**NvDataPortAnnotation**			
Note	Annotation to a port regarding a certain VariableDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.nvDataPortAnnotation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
variable	VariableDataPrototype	0..1	ref	The instance of nv data annotated. **Tags:** atp.Splitkey=variable

Table A.776: NvDataPortAnnotation

Class	**NvProvideComSpec**			
Note	Communication attributes of PPortPrototypes with respect to Nv data communication on the provided side.			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ramBlockInit Value	ValueSpecification	0..1	aggr	This represents the initial value of the RAM Block that corresponds to the referenced variable. **Tags:** atp.Splitkey=ramBlockInitValue
romBlockInit Value	ValueSpecification	0..1	aggr	This represents the initial value of the ROM block that corresponds to the referenced variable. **Tags:** atp.Splitkey=romBlockInitValue
variable	VariableDataPrototype	0..1	ref	This represents the variable for which the ComSpec is specified. **Stereotypes:** atpIdentityContributor

Table A.777: NvProvideComSpec

Class	**NvRequireComSpec**			
Note	Communication attributes of RPortPrototypes with respect to Nv data communication on the required side.			
Base	*ARObject*, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**NvRequireComSpec**			
initValue	ValueSpecification	0..1	aggr	The initial value owned by the NvComSpec **Tags:** atp.Splitkey=initValue
variable	VariableDataPrototype	0..1	ref	The VariableDataPrototype the ComSpec applies for. **Stereotypes:** atpIdentityContributor

Table A.778: NvRequireComSpec

Class	**ObdControlServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Service 08 (request control of on-board system) in relation to a particular test-Identifier (TID) supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.779: ObdControlServiceNeeds

Class	**ObdInfoServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a given InfoType (OBD Service 09) which is supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.780: ObdInfoServiceNeeds

Class	**ObdPidServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular PID (parameter identifier) which is supported by this component or module. In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.781: ObdPidServiceNeeds

Enumeration	**ObdRatioConnectionKindEnum**
Note	Defines the way how the IUMPR service connection between the Dem and the client component or module is handled (for details see the DEM Specification).
Aggregated by	ObdRatioServiceNeeds.connectionType
Literal	***Description***

▽

△

Enumeration	**ObdRatioConnectionKindEnum**
apiUse	The IUMPR service (of the DEM) uses an explicit API to connect to the component or module. **Tags:** atp.EnumerationLiteralIndex=0
observer	The IUMPR service (of the Dem) uses no API but "observes" the associated diagnostic event. **Tags:** atp.EnumerationLiteralIndex=1

Table A.782: ObdRatioConnectionKindEnum

Class	**ObdRatioServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular "ratio monitoring" which is supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connectionType	ObdRatioConnection KindEnum	0..1	attr	Defines how the DEM is connected to the component or module to perform the IUMPR (In use monitor performance ratio) service.
rateBased MonitoredEvent	DiagnosticEventNeeds	0..1	ref	The rate based monitored Diagnostic Event. **Tags:** atp.Splitkey=rateBasedMonitoredEvent
usedFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for the rate based monitor. This is an optional attribute. **Tags:** atp.Splitkey=usedFid

Table A.783: ObdRatioServiceNeeds

Class	**OffsetTimingConstraint**			
Note	Bounds the time offset between the occurrence of two timing events, without requiring a direct functional dependency between the `source` and the `target`. If the `target` event occurs, it is expected to occur earliest with the `minimum`, and latest with the `maximum` offset relatively after the occurrence of the `source` event. Note: not every `source` event occurrence shall be followed by a `target` event occurrence. In contrast to `LatencyTimingConstraint`, there shall not necessarily be a causal dependency between the `source` and `target` event.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maximum	MultidimensionalTime	0..1	aggr	The maximum offset the target event occurs relatively after the occurrence of the source event. **Tags:** atp.Splitkey=maximum xml.sequenceOffset=20
minimum	MultidimensionalTime	0..1	aggr	The mimum offset the target event occurs relatively after the occurrence of the source event. **Tags:** atp.Splitkey=minimum xml.sequenceOffset=10

▽

△

Class	OffsetTimingConstraint			
source	TimingDescriptionEvent	0..1	ref	The timing event that the target event is to be synchronized with. **Tags:** atp.Splitkey=source
target	TimingDescriptionEvent	0..1	ref	The timing event which is expected to occur timely after the source event. **Tags:** atp.Splitkey=target

Table A.784: OffsetTimingConstraint

Class	OperationInSystemInstanceRef			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ClientIdDefinition.clientServerOperation, ClientServerToSignalMapping.clientServerOperation, PortElementToCommunicationResourceMapping.clientServerOperation, ServerComSpecProps.clientServerOperation, SwcToSwcOperationArguments.operation			
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	0..1	ref	**Tags:** xml.sequenceOffset=40
targetOperation	ClientServerOperation	0..1	ref	**Tags:** xml.sequenceOffset=50

Table A.785: OperationInSystemInstanceRef

Class	OperationInvokedEvent			
Note	This event is raised when the `ClientServerOperation` referenced in `OperationInvokedEvent.operation` shall be invoked.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation which shall be invoked. **Tags:** atp.Splitkey=operation, operation.contextPPort, operation.targetProvidedOperation **InstanceRef implemented by:** POperationInAtomicSwcInstanceRef

Table A.786: OperationInvokedEvent

Class	OsTaskExecutionEvent
Note	This `RTEEvent` is supposed to execute `RunnableEntity`s which have to react on the execution of specific OsTasks. Therefore, this event is unconditionally raised whenever the OsTask on which it is mapped is executed. The main use case for this event is scheduling of `RunnableEntity`s of Complex Drivers which have to react on task executions.
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*

▽

△

Class	OsTaskExecutionEvent			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.787: OsTaskExecutionEvent

Class	OsTaskProxy			
Note	This meta-class represents a proxy for an OsTask in the System Description. **Tags:** atp.recommendedPackage=OsTaskProxies This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
period	TimeValue	0..1	attr	This attribute specifies the period in seconds of this task in case of a cyclically activated task. Please note that this attribute is informative and not directly relevant for the AUTOSAR OS. But the attribute value can be mapped into the OS configuration to support configuration work flows using a fixed set of OsTasks.
preemptability	OsTaskPreemptability Enum	0..1	attr	This attribute defines the preemptability of the task.
priority	PositiveInteger	0..1	attr	This attribute defines the priority of a task as a relative value, i.e. the values show only the relative ordering of the tasks.

Table A.788: OsTaskProxy

Class	***PPortComSpec*** (abstract)			
Note	Communication attributes of a provided `PortPrototype`. This class will contain attributes that are valid for all kinds of provide ports, independent of client-server or sender-receiver communication patterns.			
Base	*ARObject*			
Subclasses	ModeSwitchSenderComSpec, NvProvideComSpec, ParameterProvideComSpec, *SenderComSpec*, ServerComSpec			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.789: PPortComSpec

Class	PPortPrototype			
Note	Component port providing a certain port interface.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**PPortPrototype**			
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedInterface

Table A.790: PPortPrototype

Class	**PRPortPrototype**			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provided Required Interface	PortInterface	0..1	tref	This represents the `PortInterface` used to type the `PRPortPrototype`. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedRequiredInterface

Table A.791: PRPortPrototype

Class	**ParameterAccess**			
Note	The presence of a `ParameterAccess` implies that a `RunnableEntity` needs access to a `ParameterDataPrototype`.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.parameterAccess			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessed Parameter	AutosarParameterRef	0..1	aggr	Reference to the accessed calibration parameter. **Tags:** atp.Splitkey=accessedParameter
swDataDef Props	SwDataDefProps	0..1	aggr	This allows denote instance and access specific properties, mainly input values and common axis. **Tags:** atp.Splitkey=swDataDefProps

Table A.792: ParameterAccess

Class	**ParameterDataPrototype**			
Note	A `ParameterDataPrototype` represents a formalized generic piece of information that is typically immutable by the application software layer, but mutable by measurement and calibration tools. `ParameterDataPrototype` is used in various contexts and the specific context gives the otherwise generic `ParameterDataPrototype` a dedicated semantics.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswInternalBehavior.perInstanceParameter, *InternalBehavior*.constantMemory, NvBlockDescriptor.romBlock, ParameterInterface.parameter, SwcInternalBehavior.perInstanceParameter, SwcInternalBehavior.sharedParameter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype **Tags:** atp.Splitkey=initValue

Table A.793: ParameterDataPrototype

Class	**ParameterInAtomicSWCTypeInstanceRef**			
Note	This class implements an instance reference which can be applied for variables as well as for parameters.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	AutosarParameterRef.autosarParameter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	AtomicSwComponent Type	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This ist the context in a compositeDataType. **Tags:** xml.sequenceOffset=40
portPrototype	PortPrototype	0..1	ref	This is the port providing the variable or the entry point to the variable structure. **Tags:** xml.sequenceOffset=20
rootParameter DataPrototype	DataPrototype	0..1	ref	This represents the entry point for references into a CompositeDataType. **Tags:** xml.sequenceOffset=30
targetData Prototype	DataPrototype	0..1	ref	This is the target parameter element. Note that this must be nested in ParameterDataPrototype. The target must be one of ParameterDataPrototype, Application CompositeElementDataPrototype. **Tags:** xml.sequenceOffset=50

Table A.794: ParameterInAtomicSWCTypeInstanceRef

Class	**ParameterInterface**			
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	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
parameter	ParameterData Prototype	*	aggr	The ParameterDataPrototype of this ParameterInterface. **Tags:** atp.Splitkey=parameter, parameter.shortName

Table A.795: ParameterInterface

Class	**ParameterPortAnnotation**			
Note	Annotation to a port used for calibration regarding a certain ParameterDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.parameterPortAnnotation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
parameter	ParameterData Prototype	0..1	ref	The instance of annotated ParameterDataPrototype. **Tags:** atp.Splitkey=parameter

Table A.796: ParameterPortAnnotation

Class	ParameterProvideComSpec			
Note	"Communication" specification that applies to parameters on the provided side of a connection.			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	The initial value applicable for the corresponding ParameterDataPrototype. **Tags:** atp.Splitkey=initValue
parameter	ParameterData Prototype	0..1	ref	The ParameterDataPrototype to which the Parameter ComSpec applies. **Stereotypes:** atpIdentityContributor

Table A.797: ParameterProvideComSpec

Class	ParameterRequireComSpec			
Note	"Communication" specification that applies to parameters on the required side of a connection.			
Base	*ARObject*, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	The initial value applicable for the corresponding ParameterDataPrototype. **Tags:** atp.Splitkey=initValue
parameter	ParameterData Prototype	0..1	ref	The ParameterDataPrototype to which the Parameter RequireComSpec applies. **Stereotypes:** atpIdentityContributor

Table A.798: ParameterRequireComSpec

Class	ParameterSwComponentType			
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*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
constant Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular ParameterSwComponentType **Tags:** atp.Splitkey=constantMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular ParameterSwComponentType **Tags:** atp.Splitkey=dataTypeMapping

▽

△

Class	ParameterSwComponentType			
instantiation DataDefProps	InstantiationDataDefProps	*	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:** atp.Splitkey=instantiationDataDefProps, instantiationDataDefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.799: ParameterSwComponentType

Class	PassThroughSwConnector			
Note	This kind of `SwConnector` can be used inside a `CompositionSwComponentType` to connect two delegation `PortPrototype`s.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
providedOuterPort	AbstractProvidedPortPrototype	0..1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector. **Tags:** atp.Splitkey=providedOuterPort
requiredOuterPort	AbstractRequiredPortPrototype	0..1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector. **Tags:** atp.Splitkey=requiredOuterPort

Table A.800: PassThroughSwConnector

Class	*Pdu* (abstract)			
Note	Collection of all Pdus that can be routed through a bus interface.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	GeneralPurposePdu, *IPdu*, NmPdu, UserDefinedPdu			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hasDynamicLength	Boolean	0..1	attr	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].
length	UnlimitedInteger	0..1	attr	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. The Pdu length of zero bytes is allowed.

Table A.801: Pdu

Class	PduActivationRoutingGroup			
Note	Group of Pdus that can be activated or deactivated for transmission over a socket connection.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AbstractServiceInstance*.methodActivationRoutingGroup, ConsumedEventGroup.pduActivationRoutingGroup, EventHandler.pduActivationRoutingGroup			
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. **Tags:** atp.Splitkey=iPduIdentifierTcp
iPduIdentifierUdp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Udp in case that the referencing PduActivationRoutingGroup is activated. **Tags:** atp.Splitkey=iPduIdentifierUdp

Table A.802: PduActivationRoutingGroup

Class	PduMappingDefaultValue			
Note	Default Value which will be distributed if no I-Pdu has been received since last sending.			
Base	*ARObject*			
Aggregated by	TargetIPduRef.defaultValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defaultValueElement	DefaultValueElement	*	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. **Tags:** atp.Splitkey=defaultValueElement

Table A.803: PduMappingDefaultValue

Class	«atpPrototype» **PduToFrameMapping**			
Note	A PduToFrameMapping defines the composition of Pdus in each frame.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Frame*.pduToFrameMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
packingByteOrder	ByteOrderEnum	0..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	0..1	ref	Reference to a I-Pdu, N-Pdu or NmPdu that is transmitted in the Frame. **Tags:** atp.Splitkey=pdu

▽

△

Class	«atpPrototype» **PduToFrameMapping**			
startPosition	Integer	0..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 packingByte Order 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.
update IndicationBit Position	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 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 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.

Table A.804: PduToFrameMapping

Class	**PduTriggering**			
Note	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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.pduTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**PduTriggering**			
iPdu	Pdu	0..1	ref	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. 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. **Tags:** atp.Splitkey=iPdu
iPduPort	IPduPort	*	ref	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=iPduPort
iSignal Triggering	ISignalTriggering	*	ref	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.iSignal Triggering, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
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. **Tags:** atp.Splitkey=secOcCryptoMapping atp.Status=obsolete
secOcCrypto ServiceMapping	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secOcCryptoServiceMapping, secOcCrypto ServiceMapping.secOcCryptoServiceMapping, secOc CryptoServiceMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
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. **Tags:** atp.Splitkey=triggerIPduSendCondition

Table A.805: PduTriggering

Class	PerInstanceMemory			
Note	Defines a 'C' typed memory-block that needs to be available for each instance of the SW-component. This is typically only useful if supportsMultipleInstantiation is set to "true" or if the software-component defines NVRAM access via permanent blocks.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.perInstanceMemory			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	String	0..1	attr	Specifies initial value(s) of the PerInstanceMemory
swDataDef Props	SwDataDefProps	0..1	aggr	This represents the ability to allocate RAM at specific memory sections, for example, to support the RAM Block recovery strategy by mapping to uninitialized RAM. **Tags:** atp.Splitkey=swDataDefProps
type	CIdentifier	0..1	attr	The name of the "C"-type
typeDefinition	String	0..1	attr	A definition of the type with the syntax of a 'C' typedef.

Table A.806: PerInstanceMemory

Class	PerInstanceMemorySize			
Note	Resources needed by the allocation of PerInstanceMemory for each SWC instance. Note that these resources are not covered by an ObjectFileSection, because they are supposed to be allocated by the RTE.			
Base	*ARObject*			
Aggregated by	SwcImplementation.perInstanceMemorySize			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
alignment	PositiveInteger	0..1	attr	Required alignment (1,2,4,...) of the referenced Per InstanceMemory. Unit: byte.
perInstance Memory	PerInstanceMemory	0..1	ref	This represents the referenced PerInstanceMemory. **Tags:** atp.Splitkey=perInstanceMemory
size	PositiveInteger	0..1	attr	Size (in bytes) of the reference perInstanceMemory. The aggregation of PerInstanceMemorySize is subject to variability with the purpose to support variability in the software components implementations. Different algorithms in the implementation might require a different PerInstanceMemorySize. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.807: PerInstanceMemorySize

Class	PeriodicEventTriggering			
Note	Describes the behavior of an event with a strict periodic occurrence pattern, given by `period`. Additionally, it is possible to soften the strictness of the periodic occurrence behavior by specifying a `jitter`, so that there can be a deviation from the `period` up to the size of the `jitter`.			
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
jitter	MultidimensionalTime	0..1	aggr	The maximum deviation of the periodic event occurrence. **Tags:** atp.Splitkey=jitter xml.sequenceOffset=20



△

Class	**PeriodicEventTriggering**			
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The minimum time distance between subsequent consecutive occurrences of the associated event. If the `minimumInterArrivalTime` is less than the `period` minus the `jitter`, then the `minimumInterArrivalTime` has no effect on the properties of the constraint. **Tags:** atp.Splitkey=minimumInterArrivalTime xml.sequenceOffset=10
period	MultidimensionalTime	0..1	aggr	The periodic distance between subsequent occurrences of the event. **Tags:** atp.Splitkey=period xml.sequenceOffset=30

Table A.808: PeriodicEventTriggering

Class	**PermissibleSignalPath**			
Note	The PermissibleSignalPath describes the way a data element shall take in the topology. The path is described by ordered references to PhysicalChannels. 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). This has an effect on the frame generation and the frame path.			
Base	*ARObject*, *SignalPathConstraint*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
operation	SwcToSwcOperation Arguments	*	aggr	The arguments of an operation that can take the predefined way in the topology. **Tags:** atp.Splitkey=operation
physical Channel (ordered)	PhysicalChannel	*	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. **Tags:** atp.Splitkey=signal

Table A.809: PermissibleSignalPath

Class	**PhysConstrs**			
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*			
Aggregated by	DataConstrRule.physConstrs			
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
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:** atp.Splitkey=unit 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 A.810: PhysConstrs

Class	*PhysicalChannel* (abstract)			
Note	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. 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.#			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanPhysicalChannel*, EthernetPhysicalChannel, FlexrayPhysicalChannel, LinPhysicalChannel, UserDefinedPhysicalChannel			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
comm Connector	Communication Connector	*	ref	Reference to the ECUInstance via a Communication Connector to which the channel is connected. atpVariation: Variable assignment of Physical Channels to different CommunicationConnectors is expressed with this variation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commConnector, comm Connector.communicationConnector, comm Connector.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	***PhysicalChannel*** (abstract)			
frameTriggering	FrameTriggering	*	aggr	One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=frameTriggering, frameTriggering.shortName, frameTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
iSignal Triggering	ISignalTriggering	*	aggr	One ISignalTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of ISignaltriggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.shortName, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
managed Physical Channel	PhysicalChannel	*	ref	Reference between a channel with role managing channel and a channel with role managed channel. **Tags:** atp.Splitkey=managedPhysicalChannel This Attribute is only used by the AUTOSAR Classic Platform.
pduTriggering	PduTriggering	*	aggr	One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.shortName, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

Table A.811: PhysicalChannel

Class	**PhysicalDimension**
Note	This class represents a physical dimension. If the physical dimension of two units is identical, then a conversion between them is possible. The conversion between units is related to the definition of the physical dimension. Note that the equivalence of the exponents does not per se define the convertibility. For example Energy and Torque share the same exponents (Nm). Please note further the value of an exponent does not necessarily have to be an integer number. It is also possible that the value yields a rational number, e.g. to compute the square root of a given physical quantity. In this case the exponent value would be a rational number where the numerator value is 1 and the denominator value is 2. **Tags:** atp.recommendedPackage=PhysicalDimensions

▽

△

Class	PhysicalDimension			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
currentExp	Numerical	0..1	attr	This attribute represents the exponent of the physical dimension "electric current". **Tags:** xml.sequenceOffset=50
lengthExp	Numerical	0..1	attr	The exponent of the physical dimension "length". **Tags:** xml.sequenceOffset=20
luminous IntensityExp	Numerical	0..1	attr	The exponent of the physical dimension "luminous intensity". **Tags:** xml.sequenceOffset=80
massExp	Numerical	0..1	attr	The exponent of the physical dimension "mass". **Tags:** xml.sequenceOffset=30
molarAmount Exp	Numerical	0..1	attr	The exponent of the physical dimension "quantity of substance". **Tags:** xml.sequenceOffset=70
temperatureExp	Numerical	0..1	attr	The exponent of the physical dimension "temperature". **Tags:** xml.sequenceOffset=60
timeExp	Numerical	0..1	attr	The exponent of the physical dimension "time". **Tags:** xml.sequenceOffset=40

Table A.812: PhysicalDimension

Class	PhysicalDimensionMapping			
Note	This class represents a specific mapping between two PhysicalDimensions.			
Base	*ARObject*			
Aggregated by	PhysicalDimensionMappingSet.physicalDimensionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstPhysical Dimension	PhysicalDimension	0..1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping. **Tags:** atp.Splitkey=firstPhysicalDimension
secondPhysical Dimension	PhysicalDimension	0..1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping. **Tags:** atp.Splitkey=secondPhysicalDimension

Table A.813: PhysicalDimensionMapping

Class	PlcaProps			
Note	This meta-class allows to configure the PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the CouplingPort (PHY).			
Base	*ARObject*			
Aggregated by	CouplingPort.plcaProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
plcaLocalNode Id	PositiveInteger	0..1	attr	This attribute defines the node ID when the PLCA mode for 10BASE-T1S is used.

▽

△

Class	**PlcaProps**			
plcaMaxBurst Count	PositiveInteger	0..1	attr	Defines maximum packets allowed to be transmitted within a TO. This configuration can be different from one ECU to another within the PLCA mixed segment.
plcaMaxBurst Timer	PositiveInteger	0..1	attr	Limits the burst frames in bit time. This configuration can be different from one ECU to another within the PLCA mixed segment. For PLCA burst mode to work properly this timer should be set greater than one IPG.

Table A.814: PlcaProps

Enumeration	**PncGatewayTypeEnum**
Note	Defines the PncGateway roles.
Aggregated by	*CommunicationConnector*.pncGatewayType
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 A.815: PncGatewayTypeEnum

Class	**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*			
Aggregated by	SystemMapping.pncMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
dynamicPnc MappingPdu Group	ISignalIPduGroup	*	ref	Reference to an ISignalIPduGroup that allows mapping of this PNC without statically mapping this PNC directly to a channel. This is needed to describe dynamic PNCs that can be learned only at run-time and which have also a relation to an ISignalIPduGroup. **Tags:** atp.Splitkey=dynamicPncMappingPduGroup atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ident	PncMappingIdent	0..1	aggr	This adds the ability to become referrable to PncMapping. **Stereotypes:** atpIdentityContributor
physical Channel	PhysicalChannel	*	ref	This reference maps the partial network to a communication channel. **Tags:** atp.Splitkey=physicalChannel

▽

△

Class	**PncMapping**			
pncConsumed ProvidedServiceInstance 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:** atp.Splitkey=pncConsumedProvidedServiceInstance Group, pncConsumedProvidedServiceInstance Group.consumedProvidedServiceInstanceGroup, pnc ConsumedProvidedServiceInstanceGroup.variation Point.shortLabel 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. **Tags:** atp.Splitkey=pncGroup This Attribute is only used by the AUTOSAR Classic Platform.
pncIdentifier	PositiveInteger	0..1	attr	Identifer of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu. **Stereotypes:** atpIdentityContributor
pncPdurGroup	PdurIPduGroup	*	ref	This reference maps the Partial Network Cluster to a set of PdurIpduGroups. **Tags:** atp.Splitkey=pncPdurGroup This Attribute is only used by the AUTOSAR Classic Platform.
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. **Tags:** atp.Status=obsolete
relevantFor DynamicPnc Mapping	EcuInstance	*	ref	Reference to a PNC Gateway ECU for PNCs which do not have a static channel mapping. This is needed to describe dynamic PNCs that can be learned only at run-time and which have no relation to an ISignalIPdu Group. **Tags:** atp.Splitkey=relevantForDynamicPncMapping atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the PncMapping. It shall be unique in the System scope. **Stereotypes:** atpIdentityContributor

▽

△

Class	**PncMapping**			
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. **Tags:** atp.Splitkey=vfc, vfc.contextComposition, vfc.contextComponent, vfc.target **InstanceRef implemented by:** PortGroupInSystem InstanceRef
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. **Tags:** atp.Splitkey=wakeupFrame This Attribute is only used by the AUTOSAR Classic Platform.

Table A.816: PncMapping

Class	**PncMappingIdent**			
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*			
Aggregated by	PncMapping.ident			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.817: PncMappingIdent

Class	**PortAPIOption**			
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 Runnable Entity to the PortPrototype).			
Base	*ARObject*			
Aggregated by	SwcInternalBehavior.portAPIOption			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
enableTake Address	Boolean	0..1	attr	If set to true, the software-component is able to use the API reference for deriving a pointer to an object.
errorHandling	DataTransformation ErrorHandlingEnum	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	0..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	0..1	ref	The option is valid for generated functions related to communication over this port **Stereotypes:** atpIdentityContributor

▽

△

Class	**PortAPIOption**			
portArgValue (ordered)	PortDefinedArgument Value	*	aggr	An argument value defined by this port.
supported Feature	SwcSupportedFeature	*	aggr	This collection specifies which features are supported by the RunnableEntitys which access a PortPrototype that it referenced by this PortAPIOption. **Tags:** atp.Splitkey=supportedFeature
transformer Status Forwarding	DataTransformation StatusForwardingEnum	0..1	attr	This attribute specifies whether a RunnableEntity accessing a PortPrototype that is referenced by this Port APIOption shall be able to forward a status code to the transformer chain.

Table A.818: PortAPIOption

Class	**PortDefinedArgumentValue**			
Note	A PortDefinedArgumentValue is passed to a RunnableEntity dealing with the ClientServerOperations provided by a given PortPrototype. Note that this is restricted to PPortPrototypes of a ClientServer Interface.			
Base	*ARObject*			
Aggregated by	PortAPIOption.portArgValue			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
value	ValueSpecification	0..1	aggr	Specifies the actual value.
valueType	ImplementationData Type	0..1	tref	The implementation type of this argument value. It should not be composite type or a pointer. **Stereotypes:** isOfType

Table A.819: PortDefinedArgumentValue

Class	**PortElementToCommunicationResourceMapping**			
Note	This meta class maps a communication resource to CP Software Clusters. In this case the kind of Port Prototype specified whether the Software Cluster has to provide or to require the resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.portElementToComResourceMapping, SystemMapping.portElementTo ComResourceMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
clientServer Operation	ClientServerOperation	0..1	iref	ClientServerOperation instance qualifying the communication resource **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef
communication Resource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=communicationResource

▽

△

Class	**PortElementToCommunicationResourceMapping**			
mode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	iref	ModeDeclarationGroupPrototype instance qualifying the communication resource **Tags:** atp.Splitkey=modeDeclarationGroupPrototype, modeDeclarationGroupPrototype.contextComposition, modeDeclarationGroupPrototype.contextComponent, modeDeclarationGroupPrototype.contextPort, mode DeclarationGroupPrototype.targetModeDeclarationGroup Prototype **InstanceRef implemented by:** ModeDeclarationGroup PrototypeInSystemInstanceRef
parameterData Prototype	ParameterData Prototype	0..1	iref	ParameterDataPrototype instance qualifying the communication resource. **Tags:** atp.Splitkey=parameterDataPrototype, parameter DataPrototype.contextComposition, parameterData Prototype.contextComponent, parameterData Prototype.contextPort, parameterDataPrototype.target ParameterDataPrototype **InstanceRef implemented by:** ParameterDataPrototype InSystemInstanceRef
trigger	Trigger	0..1	iref	Trigger instance qualifying the communication resource. **Tags:** atp.Splitkey=trigger, trigger.contextComposition, trigger.contextComponent, trigger.contextPort, trigger.targetTrigger **InstanceRef implemented by:** TriggerInSystemInstance Ref
variableData Prototype	VariableDataPrototype	0..1	iref	VariableDataPrototype instance qualifying the communication resource **Tags:** atp.Splitkey=variableDataPrototype, variableData Prototype.contextComponent, variableData Prototype.contextComposition, variableData Prototype.contextPort, variableDataPrototype.targetData Prototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef

Table A.820: PortElementToCommunicationResourceMapping

Class	**PortGroup**			
Note	Group of ports which share a common functionality. Example: 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	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.portGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**PortGroup**			
innerGroup	PortGroup	*	iref	Links a PortGroup in a composition to another PortGroup, that is defined in a component which is part of this CompositionSwComponentType. **Tags:** atp.Splitkey=innerGroup, innerGroup.context, innerGroup.target **InstanceRef implemented by:** InnerPortGroupIn CompositionInstanceRef
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:** atp.Splitkey=outerPort, outerPort.portPrototype, outerPort.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.821: PortGroup

Class	***PortInterface*** (abstract)			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ClientServerInterface, *DataInterface*, ModeSwitchInterface, TriggerInterface			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
isService	Boolean	0..1	attr	This flag is set if the `PortInterface` is to be used for communication between an • `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. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.822: PortInterface

Class	***PortInterfaceMapping*** (abstract)
Note	Specifies one `PortInterfaceMapping` to support the connection of Ports typed by two different `PortInterface`s with PortInterface elements having unequal names and/or unequal semantic (resolution or range).

▽

△

Class	**PortInterfaceMapping** (abstract)			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ClientServerInterfaceMapping, ModeInterfaceMapping, TriggerInterfaceMapping, VariableAndParameterInterfaceMapping			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.823: PortInterfaceMapping

Class	**PortPrototype** (abstract)			
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*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this `PortPrototype` with respect to client/server communication. **Tags:** atp.Splitkey=clientServerAnnotation
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port. **Tags:** atp.Splitkey=delegatedPortAnnotation
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port. **Tags:** atp.Splitkey=ioHwAbstractionServerAnnotation
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port. **Tags:** atp.Splitkey=modePortAnnotation
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port. **Tags:** atp.Splitkey=nvDataPortAnnotation
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port. **Tags:** atp.Splitkey=parameterPortAnnotation
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication. **Tags:** atp.Splitkey=senderReceiverAnnotation
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port. **Tags:** atp.Splitkey=triggerPortAnnotation

Table A.824: PortPrototype

Class	**PortPrototypeBlueprint**
Note	This meta-class represents the ability to express a blueprint of a PortPrototype by referring to a particular PortInterface. This blueprint can then be used as a guidance to create particular PortPrototypes which are defined according to this blueprint. By this it is possible to standardize application interfaces without the need to also standardize software-components with PortPrototypes typed by the standardized Port Interfaces. **Tags:** atp.recommendedPackage=PortPrototypeBlueprints

▽

△

Class	PortPrototypeBlueprint			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	PortPrototypeBlueprint InitValue	*	aggr	This specifies the init values for the dataElements in the particular PortPrototypeBlueprint. **Tags:** atp.Splitkey=initValue
interface	PortInterface	0..1	ref	This is the interface for which the blueprint is defined. It may be a blueprint itself or a standardized PortInterface **Tags:** atp.Splitkey=interface
providedComSpec	PPortComSpec	*	aggr	Provided communication attributes per interface element (data element or operation). **Tags:** atp.Splitkey=providedComSpec, providedComSpec.dataElement, providedComSpec.getter, providedComSpec.modeGroup, providedComSpec.operation, providedComSpec.parameter, providedComSpec.setter, providedComSpec.variable
requiredComSpec	RPortComSpec	*	aggr	Required communication attributes, one for each interface element. **Tags:** atp.Splitkey=requiredComSpec, requiredComSpec.dataElement, requiredComSpec.getter, requiredComSpec.modeGroup, requiredComSpec.operation, requiredComSpec.parameter, requiredComSpec.setter, requiredComSpec.shortLabel, requiredComSpec.variable

Table A.825: PortPrototypeBlueprint

Primitive	PositiveUnlimitedInteger
Note	This is a positive unlimited integer which can be denoted in decimal, binary, octal and hexadecimal. **Tags:** xml.xsd.customType=POSITIVE-UNLIMITED-INTEGER xml.xsd.pattern=0|[\+]?[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[bB][0-1]+|0[0-7]+ xml.xsd.type=string

Table A.826: PositiveUnlimitedInteger

Class	PostBuildVariantCondition			
Note	This class specifies the value which shall be assigned to a particular variant criterion in order to bind the variation point. If multiple criterion/value pairs are specified, they shall all match to bind the variation point. In other words binding can be represented by (criterion1 == value1) && (condition2 == value2) ...			
Base	*ARObject*			
Aggregated by	VariationPoint.postBuildVariantCondition, VariationPointProxy.postBuildVariantCondition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
matchingCriterion	PostBuildVariantCriterion	0..1	ref	This is the criterion which needs to match the value in order to make the PostbuildVariantCondition to be true. **Tags:** atp.Splitkey=matchingCriterion

▽

△

Class	**PostBuildVariantCondition**			
value	Integer	0..1	attr	This is the particular value of the post-build variant criterion. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.827: PostBuildVariantCondition

Class	**PostBuildVariantCriterion**			
Note	This class specifies one particular PostBuildVariantSelector. **Tags:** atp.recommendedPackage=PostBuildVariantCriterions			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuMethod	CompuMethod	0..1	ref	The compuMethod specifies the possible values for the variant criterion serving as an enumerator. **Tags:** atp.Splitkey=compuMethod

Table A.828: PostBuildVariantCriterion

Class	**PostBuildVariantCriterionValue**			
Note	This class specifies the value which shall be assigned to a particular variant criterion in order to bind the variation point. If multiple criterion/value pairs are specified, they all shall match to bind the variation point.			
Base	*ARObject*			
Aggregated by	PostBuildVariantCriterionValueSet.postBuildVariantCriterionValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
annotation	Annotation	*	aggr	This provides the ability to add information why the value is set like it is. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=30
value	Integer	0..1	attr	This is the particular value of the post-build variant criterion. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
variantCriterion	PostBuildVariant Criterion	0..1	ref	This association selects the variant criterion whose value is specified. **Tags:** atp.Splitkey=variantCriterion xml.sequenceOffset=10

Table A.829: PostBuildVariantCriterionValue

Class	PredefinedVariant			
Note	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. **Tags:** atp.recommendedPackage=PredefinedVariants			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
includedVariant	PredefinedVariant	*	ref	The associated variants are considered part of this PredefinedVariant. This means the settings of the included variants are included in the settings of the referencing PredefinedVariant. Nevertheless the included variants might be included in several predefined variants. **Tags:** atp.Splitkey=includedVariant
postBuildVariant CriterionValue Set	PostBuildVariant CriterionValueSet	*	ref	This is the postBuildVariantCriterionValueSet contributing to the predefinded variant. **Tags:** atp.Splitkey=postBuildVariantCriterionValueSet
sw Systemconstant ValueSet	SwSystemconstant ValueSet	*	ref	This ist the set of Systemconstant Values contributing to the predefined variant. **Tags:** atp.Splitkey=swSystemconstantValueSet

Table A.830: PredefinedVariant

Class	ProvidedServiceInstance			
Note	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allowedService Consumer	NetworkEndpoint	*	ref	NetworkEndpoints on which the ConsumedService Instances that are communicating with this Provided ServiceInstance are allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=allowedServiceConsumer, allowedService Consumer.networkEndpoint, allowedService Consumer.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
autoAvailable	Boolean	0..1	attr	Defines that this ProvidedServiceInstance shall be offered by the service discovery at ECU start.
eventHandler	EventHandler	*	aggr	Collection of event groups provided by the Provided ServiceInstance **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventHandler, eventHandler.shortName, eventHandler.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.
loadBalancing Priority	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**			
loadBalancing Weight	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.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the PSI is provided (udp, tcp or both). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel 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.
remoteMulticast Subscription Address	ApplicationEndpoint	*	ref	This reference defines the remote multicast subscribed 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:** atp.Splitkey=remoteMulticastSubscriptionAddress, remoteMulticastSubscriptionAddress.applicationEndpoint, remoteMulticastSubscriptionAddress.variationPoint.short Label vh.latestBindingTime=postBuild
remoteUnicast Address	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:** atp.Splitkey=remoteUnicastAddress, remoteUnicast Address.applicationEndpoint, remoteUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdServerTimer Config	SomeipSdServer ServiceInstanceConfig	0..1	ref	Server specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdServerTimerConfig, sdServerTimer Config.someipSdServerServiceInstanceConfig, sdServer TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is offered.

Table A.831: ProvidedServiceInstance

Class	**QueuedReceiverComSpec**			
Note	Communication attributes specific to queued receiving.			
Base	*ARObject*, *RPortComSpec*, *ReceiverComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**QueuedReceiverComSpec**			
queueLength	PositiveInteger	0..1	attr	Length of queue for received events. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.832: QueuedReceiverComSpec

Class	**QueuedSenderComSpec**			
Note	Communication attributes specific to distribution of events (`PPortPrototype`, `SenderReceiverInterface` and `dataElement` carries an "event").			
Base	*ARObject*, *PPortComSpec*, *SenderComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.833: QueuedSenderComSpec

Class	***RPortComSpec*** (abstract)			
Note	Communication attributes of a required PortPrototype. This class will contain attributes that are valid for all kinds of require-ports, independent of client-server or sender-receiver communication patterns.			
Base	*ARObject*			
Subclasses	ClientComSpec, ModeSwitchReceiverComSpec, NvRequireComSpec, ParameterRequireComSpec, *ReceiverComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.834: RPortComSpec

Class	**RPortPrototype**			
Note	Component port requiring a certain port interface.			
Base	*ARObject*, *AbstractRequiredPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mayBe Unconnected	Boolean	0..1	attr	If set to true, this attribute indicates that the enclosing `RPortPrototype` may be left unconnected and that this aspect has explicitly been considered in the software-component's design. This Attribute is only used by the AUTOSAR Classic Platform.
required Interface	PortInterface	0..1	tref	The interface that this port requires. **Stereotypes:** isOfType **Tags:** atp.Splitkey=requiredInterface

Table A.835: RPortPrototype

Class	***RTEEvent*** (abstract)			
Note	Abstract base class for all RTE-related events			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Subclasses	AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, OsTaskExecutionEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, TimingEvent, TransformerHardErrorEvent			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. **Tags:** atp.Splitkey=disabledMode, disabledMode.contextPort, disabledMode.contextModeDeclarationGroupPrototype, disabledMode.targetModeDeclaration **InstanceRef implemented by:** RModeInAtomicSwcInstanceRef
startOnEvent	RunnableEntity	0..1	ref	The referenced `RunnableEntity` starts when the corresponding `RTEEvent` is raised. **Tags:** atp.Splitkey=startOnEvent

Table A.836: RTEEvent

Class	**RapidPrototypingScenario**			
Note	This meta-class provides the ability to describe a Rapid Prototyping Scenario. Such a Rapid Prototyping Scenario consist out of two main aspects, the description of the byPassPoints and the relation to an rpt Hook. **Tags:** atp.recommendedPackage=RapidPrototypingScenarios			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hostSystem	System	0..1	ref	System which describes the software components of the host ECU. **Tags:** atp.Splitkey=hostSystem
rptContainer	RptContainer	*	aggr	Top-level rptContainer definitions of this specific rapid prototyping scenario. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptContainer, rptContainer.shortName, rpt Container.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptProfile	RptProfile	*	aggr	Defiens the applicable Rapid Prototyping profils which are especially defining the smbol of the service functions and the valid id range. The order of the RptProfiles determines the order of the service function invocation by RTE. **Tags:** atp.Splitkey=rptProfile, rptProfile.shortName atp.Status=obsolete
rptSystem	System	0..1	ref	System which describes the rapid prototyping algorithm in the format of AUTOSAR Software Components. **Tags:** atp.Splitkey=rptSystem

Table A.837: RapidPrototypingScenario

Class	***ReceiverComSpec*** (abstract)			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	*ARObject*, *RPortComSpec*			
Subclasses	NonqueuedReceiverComSpec, QueuedReceiverComSpec			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compositeNetworkRepresentation	CompositeNetworkRepresentation	*	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. **Tags:** atp.Splitkey=compositeNetworkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
dataElement	AutosarDataPrototype	0..1	ref	Data element these attributes belong to. **Stereotypes:** atpIdentityContributor
handleOutOfRange	HandleOutOfRangeEnum	0..1	attr	This attribute controls how values that are out of the specified range are handled according to the values of HandleOutOfRangeEnum. This Attribute is only used by the AUTOSAR Classic Platform.
handleOutOfRangeStatus	HandleOutOfRangeStatusEnum	0..1	attr	Control the way how return values are created in case of an out-of-range situation. This Attribute is only used by the AUTOSAR Classic Platform.
networkRepresentation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus. **Tags:** atp.Splitkey=networkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
receptionProps	ReceptionComSpecProps	0..1	aggr	"This aggregation represents the definition transmission props in the context of the enclosing ReceiverComSpec. **Tags:** atp.Splitkey=receptionProps
replaceWith	VariableAccess	0..1	aggr	This aggregation is used to identify the AutosarDataPrototype to be taken for sourcing an external replacement in the out-of-range and invalidValue handling. **Tags:** atp.Splitkey=replaceWith, replaceWith.shortName This Attribute is only used by the AUTOSAR Classic Platform.
transformationComSpecProps	TransformationComSpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation. **Tags:** atp.Splitkey=transformationComSpecProps

Table A.838: ReceiverComSpec

Class	**ReceptionComSpecProps**			
Note	This meta-class defines a set of reception attributes which the application software is assumed to implement.			
Base	*ARObject*			
Aggregated by	*ReceiverComSpec*.receptionProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ReceptionComSpecProps**			
comHandler TaskMapping Enabled	Boolean	0..1	attr	This attribute defines whether the ComHandler Task Mapping is activated or deactivated. This Attribute is only used by the AUTOSAR Classic Platform.
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, but may also indicate a general data reception period.
invalidValue BitfieldErrors Enabled	Boolean	0..1	attr	This attribute defines whether the invalid value error bitfield shall be used which contains the information of invalid value errors of the direct child elements of the enclosing structure and not of elements of sub-structures. This Attribute is only used by the AUTOSAR Classic Platform.
outOfRange BitfieldErrors Enabled	Boolean	0..1	attr	This attribute defines whether the out-of-range error bitfield shall be used which contains the information of out-of-range errors of the direct child elements of the enclosing structure and not of elements of sub-structures. This Attribute is only used by the AUTOSAR Classic Platform.
timeout	TimeValue	0..1	attr	This attribute defines the time interval after which the application shall assume that the to be received data reception has timed out, i.e. the respective data has not been received for that amount of time.

Table A.839: ReceptionComSpecProps

Class	**RecordValueSpecification**			
Note	Specifies the values for a record.			
Base	*ARObject*, *CompositeValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.argument, Constant Specification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compare Value, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compare Value, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
field (ordered)	ValueSpecification	*	aggr	The value for a single record field. This could also be mapped explicitly to a record element of the data type using the shortName of the ValueSpecification. But this would introduce a relationship to the data type that is too strong. As of now, it is only important that the structure of the data type matches the structure of the Value Specification independently of the shortNames. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.840: RecordValueSpecification

Enumeration	**ReentrancyLevelEnum**
Note	Specifies if and in which kinds of environments an entity is reentrant.
Aggregated by	*ExecutableEntity*.reentrancyLevel
Literal	***Description***
multicoreReentrant	Unlimited concurrent execution of this entity is possible, including preemption and parallel execution on multi core systems. **Tags:** atp.EnumerationLiteralIndex=0
nonReentrant	Concurrent execution of this entity is not possible. **Tags:** atp.EnumerationLiteralIndex=1
singleCore Reentrant	Pseudo-concurrent execution (i.e. preemption) of this entity is possible on single core systems. **Tags:** atp.EnumerationLiteralIndex=2

Table A.841: ReentrancyLevelEnum

Primitive	**Ref**			
Note	This primitive denotes a name based reference. For detailed syntax see the xsd.pattern. • first slash (relative or absolute reference) [optional] • Identifier [required] • a sequence of slashes and Identifiers [optional] This primitive is used by the meta-model tools to create the references. **Tags:** xml.xsd.customType=REF xml.xsd.pattern=/?[a-zA-Z][a-zA-Z0-9_]{0,127}(/[a-zA-Z][a-zA-Z0-9_]{0,127})* xml.xsd.type=string			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	Identifier	0..1	attr	This attribute reflects the base to be used for this reference. **Tags:** xml.attribute=true
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=valid xml.attribute=true
index	PositiveInteger	0..1	attr	This attribute supports the use case to point on specific elements in an array. This is in particular required if arrays are used to implement particular data objects. The counting of array indices starts with the value 0, i.e. the index of the first array element is 0. **Tags:** xml.attribute=true

Table A.842: Ref

Class	**ReferenceBase**			
Note	This meta-class establishes a basis for relative references. Reference bases are identified by the short Label which shall be unique in the current package.			
Base	*ARObject*			
Aggregated by	ARPackage.referenceBase			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ReferenceBase**			
globalElement	ReferrableSubtypes Enum	*	attr	This attribute represents a meta-class for which the global referencing is supported via this reference base. **Tags:** xml.sequenceOffset=29
globalIn Package	ARPackage	*	ref	This represents the ability to express that global elements live in various packages which do not have a common ancestor package. Packages mentioned by Reference Base.globalInPackage are used in addition to the one in ReferenceBase.package. **Tags:** atp.Splitkey=globalInPackage xml.sequenceOffset=28
package	ARPackage	0..1	ref	This association specifies the basis of all relative references with the base equals shortLabel. **Tags:** atp.Splitkey=package xml.sequenceOffset=30
shortLabel	Identifier	0..1	attr	This is the name of the reference base. By this name, particular references can denote the applicable base. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10

Table A.843: ReferenceBase

Class	**ReferenceValueSpecification**			
Note	Specifies a reference to a data prototype to be used as an initial value for a pointer in the software.			
Base	*ARObject*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
referenceValue	DataPrototype	0..1	ref	The referenced data prototype. **Tags:** atp.Splitkey=referenceValue

Table A.844: ReferenceValueSpecification

Class	***Referrable*** (abstract)
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).
Base	*ARObject*
Subclasses	*AtpDefinition*, BswDistinguishedPartition, *BswModuleCallPoint*, BswModuleClientServerEntry, Bsw VariableAccess, CouplingPortTrafficClassAssignment, *DiagnosticEnvModeElement*, EthernetPriority Regeneration, ExclusiveAreaNestingOrder, *HwDescriptionEntity*, *ImplementationProps*, LinSlaveConfig Ident, ModeTransition, *MultilanguageReferrable*, PncMappingIdent, *SingleLanguageReferrable*, SoCon IPduIdentifier, TpConnectionIdent

▽

△

Class	***Referrable*** (abstract)			
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. **Stereotypes:** atpIdentityContributor **Tags:** xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:** atp.Splitkey=shortNameFragment xml.sequenceOffset=-90

Table A.845: Referrable

Class	**RelativeTolerance**			
Note	Maximum allowable deviation			
Base	*ARObject*, *TimeRangeTypeTolerance*			
Aggregated by	TimeRangeType.tolerance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
relative	Integer	0..1	attr	Maximum allowable deviation in percent (percent of the corresponding TimeValue).

Table A.846: RelativeTolerance

Class	**RequestResponseDelay**			
Note	Time to wait before answering the query.			
Base	*ARObject*			
Aggregated by	SomeipSdClientEventGroupTimingConfig.requestResponseDelay, SomeipSdServerEventGroupTimingConfig.requestResponseDelay, SomeipSdServerServiceInstanceConfig.requestResponseDelay			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxValue	TimeValue	0..1	attr	Maximum allowable response delay to entries received by multicast in seconds.
minValue	TimeValue	0..1	attr	Minimum allowable response delay to entries received by multicast in seconds.

Table A.847: RequestResponseDelay

Class	**RoleBasedBswModuleEntryAssignment**			
Note	This class specifies an assignment of a role to a particular BswModuleEntry (usually a configurable callback). With this assignment, the role of the callback is mapped to a specific ServiceNeeds element, so that a tool is able to create appropriate configuration values for the module that implements the AUTOSAR Service.			
Base	*ARObject*			
Aggregated by	BswServiceDependency.assignedEntryRole			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	RoleBasedBswModuleEntryAssignment			
assignedEntry	BswModuleEntry	0..1	ref	The assigned entry. It should be an implementedEntry or expectedEntry of the module or cluster that requires the ServiceNeeds. **Tags:** atp.Splitkey=assignedEntry This Attribute is only used by the AUTOSAR Classic Platform.
role	Identifier	0..1	attr	This is the role of the assigned BswModuleEntry in the given context. The attribute is required (for example) because different kind of callbacks may be associated with the same ServiceNeeds (e.g. end-notification vs. error-notification). The value shall be the role name of a configurable function call (usually a callback) as standardized in the Software Specification of the related AUTOSAR Service.

Table A.848: RoleBasedBswModuleEntryAssignment

Class	RoleBasedDataAssignment			
Note	This class specifies an assignment of a role to a particular data object in either • the SwcInternalBehavior of a software component (or in the BswInternalBehavior of a BSW module or BSW cluster) in the context of an AUTOSAR Service or • an NvBlockDescriptor to sort out the assignment of event-based writing strategies to data elements in a PortPrototype. With this assignment, the role of the data can be mapped to a DataPrototype that is used in the context of the definition of a specific ServiceNeeds or NvBlockDescriptor, so that a tool is able to create the correct access or writing strategy.			
Base	*ARObject*			
Aggregated by	BswServiceDependency.assignedData, NvBlockDescriptor.writingStrategy, SwcService Dependency.assignedData			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
role	Identifier	0..1	attr	This is the role of the assigned data in the given context. Possible values need to be specified on M1 level. Additionally the TPS Software Component Template provides a list of applicable roles for various service dependencies and service use cases in chapter 13 "Service Dependencies and Service Use Cases" (e.g., ramBlock in case of the needs for a permanent RAM block).
usedData Element	AutosarVariableRef	0..1	aggr	The VariableDataPrototype used in this role, e.g. • Permanent RAM Block of an NVRAM Block which shall belong to the same SwcInternalBehavior or Bsw InternalBehavior. • In the role signalBasedDiagnostics it has to refer to a VariableDataPrototype in a SenderReceiverInterface or a NvDataInterface. **Tags:** atp.Splitkey=usedDataElement
usedParameter Element	AutosarParameterRef	0..1	aggr	The ParameterDataPrototype used in this role, e.g. • ROM Block of an NVRAM Block. It shall belong to the same SwcInternalBehavior or BswInternalbehavior. • In the role signalBasedDiagnostics it has to refer to a ParameterDataPrototype in a ParameterInterface. **Tags:** atp.Splitkey=usedParameterElement

▽

△

Class	RoleBasedDataAssignment			
usedPim	PerInstanceMemory	0..1	ref	The (untyped) PerInstanceMemory used in this role (e.g. as a Permanent RAM Block for an NVRAM Block). **Tags:** atp.Splitkey=usedPim

Table A.849: RoleBasedDataAssignment

Class	RoleBasedDataTypeAssignment			
Note	This class specifies an assignment of a role to a particular data type of a software component (or in the BswModuleBehavior of a module or cluster) in the context of an AUTOSAR Service. With this assignment, the role of the data type can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct access.			
Base	*ARObject*			
Aggregated by	*ServiceDependency*.assignedDataType			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
role	Identifier	0..1	attr	This is the role of the associated data type in the given context.
used Implementation DataType	ImplementationData Type	0..1	ref	This represents the associated ImplementationDataType. **Tags:** atp.Splitkey=usedImplementationDataType

Table A.850: RoleBasedDataTypeAssignment

Class	RoleBasedPortAssignment			
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*			
Aggregated by	NvBlockDescriptor.clientServerPort, SwcServiceDependency.assignedPort			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
portPrototype	PortPrototype	0..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. **Tags:** atp.Splitkey=portPrototype
role	Identifier	0..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 A.851: RoleBasedPortAssignment

Class	RootSwCompositionPrototype			
Note	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 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. 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. The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, System.rootSoftwareComposition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
calibration ParameterValue Set	CalibrationParameter ValueSet	*	ref	Used CalibrationParameterValueSet for instance specific initialization of calibration parameters. **Tags:** atp.Splitkey=calibrationParameterValueSet This Attribute is only used by the AUTOSAR Classic Platform.
flatMap	FlatMap	0..1	ref	The FlatMap used in the scope of this RootSw CompositionPrototype. **Tags:** atp.Splitkey=flatMap This Attribute is only used by the AUTOSAR Classic Platform.
software Composition	CompositionSw ComponentType	0..1	tref	We assume that there is exactly one top-level composition that includes all Component instances of the system. **Stereotypes:** isOfType **Tags:** atp.Splitkey=softwareComposition

Table A.852: RootSwCompositionPrototype

Class	RoughEstimateHeapUsage			
Note	Rough estimation of the heap usage.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memory Consumption	PositiveInteger	0..1	attr	Rough estimate of the heap usage. Unit: byte.

Table A.853: RoughEstimateHeapUsage

Class	RoughEstimateOfExecutionTime			
Note	Provides a description of a rough estimate on the ExecutionTime.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
additional Information	String	0..1	attr	Provides description on the rough estimate of the ExecutionTime.
estimated ExecutionTime	MultidimensionalTime	0..1	aggr	The estimated execution time. **Tags:** atp.Splitkey=estimatedExecutionTime

Table A.854: RoughEstimateOfExecutionTime

Class	**RoughEstimateStackUsage**			
Note	Rough estimation of the stack usage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
memory Consumption	PositiveInteger	0..1	attr	Rough estimate of the stack usage. Unit: byte.

Table A.855: RoughEstimateStackUsage

Class	**RptComponent**			
Note	Description of component instance for which rapid prototyping support is implemented.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RptSupportData.rptComponent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mcData Assignment	RoleBasedMcData Assignment	*	aggr	Reference to related McDataElement describing the implementation of "RP global buffer", "RP global measurement buffer", "RP enabler flag" and the "RP runnable disabler flag". **Tags:** atp.Splitkey=mcDataAssignment
rpImplPolicy	RptImplPolicy	0..1	aggr	Describes the implemented code preparation for rapid prototyping at data accesses. **Tags:** atp.Splitkey=rpImplPolicy atp.Status=obsolete
rptExecutable Entity	RptExecutableEntity	*	aggr	ExecutableEntity instance which can be bypassed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptExecutableEntity, rptExecutable Entity.shortName, rptExecutableEntity.variationPoint.short Label vh.latestBindingTime=preCompileTime

Table A.856: RptComponent

Class	**RptContainer**			
Note	This meta-class defines a byPassPoint and the relation to a rptHook. Additionally it may contain further rptContainers if the byPassPoint is not atomic. For example a byPass Point referencing to a RunnableEntity may contain rptContainers referring to the data access points of the RunnableEntity. The RptContainer structure on M1 shall follow the M1 structure of the Software Component Descriptions. The category attribute denotes which level of the Software Component Description is annotated.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RapidPrototypingScenario.rptContainer, RptContainer.rptContainer			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**RptContainer**			
byPassPoint	AtpFeature	*	iref	byPassPoint describes the required preparation of the host ECU. At a byPassPoint the host ECU shall be capable to communicate with a RPT System in order to support the execution of the rapid prototyping algorithms with the original data calculated by the host system and to replace dedicated results of the host system by the results of the rapid prototyping algorithm. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=byPassPoint, byPassPoint.contextElement, byPassPoint.target, byPassPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime **InstanceRef implemented by:** AnyInstanceRef
explicitRpt ProfileSelection	RptProfile	*	ref	This attribute defines the applicable RptProfiles for the specific RptContainer. If not any references to a specific RptProfile is defined, all RptProfiles defined in the Rapid PrototypingScenario are applicable. **Tags:** atp.Splitkey=explicitRptProfileSelection atp.Status=obsolete
rptContainer	RptContainer	*	aggr	Sub-level rptContainer definitions of this specific rapid prototyping scenario. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptContainer, rptContainer.shortName, rpt Container.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptExecutable EntityProperties	RptExecutableEntity Properties	0..1	aggr	Describes the required code preparation for rapid prototyping at ExecutableEntity invocation. **Tags:** atp.Splitkey=rptExecutableEntityProperties atp.Status=obsolete
rptHook	RptHook	0..1	aggr	The rptHook describes the link between a byPassPoint and the rapid prototyping algorithm. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptHook, rptHook.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the required code preparation for rapid prototyping at data accesses. **Tags:** atp.Splitkey=rptImplPolicy atp.Status=obsolete
rptSw Prototyping Access	RptSwPrototyping Access	0..1	aggr	Describes the required accessibility of data and modes by the rapid prototyping tooling. **Tags:** atp.Splitkey=rptSwPrototypingAccess atp.Status=obsolete

Table A.857: RptContainer

Class	**RptExecutableEntity**
Note	This describes a ExecutableEntity instance which can be bypassed.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	RptComponent.rptExecutableEntity

▽

△

Class	**RptExecutableEntity**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
rptExecutable EntityEvent	RptExecutableEntity Event	*	aggr	ExecutableEntity event instance activation the owning Rpt ExecutableEntity. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptExecutableEntityEvent, rptExecutable EntityEvent.shortName, rptExecutableEntity Event.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptRead	RoleBasedMcData Assignment	*	aggr	read access to a variable **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptRead, rptRead.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptWrite	RoleBasedMcData Assignment	*	aggr	write access to a variable **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptWrite, rptWrite.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbol	CIdentifier	0..1	attr	The symbol describing this ExecutableEntity's entry point.

Table A.858: RptExecutableEntity

Class	**RptExecutableEntityEvent**			
Note	This describes an ExecutableEntity event instance which can be bypassed.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RptExecutableEntity.rptExecutableEntityEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
execution Context	RptExecutionContext	*	ref	This describes the context in which the event of the executable entity is executed. **Tags:** atp.Splitkey=executionContext
mcData Assignment	RoleBasedMcData Assignment	*	aggr	Reference to related McDataElements describing the implementation of "RP runnable disabler flag" and "stimulation enabler flag" The possible roles of the RoleBasedMcData Assignment.role attribute are: • RpRunnableDisablerFlag **Tags:** atp.Splitkey=mcDataAssignment
rptEventId	PositiveInteger	0..1	attr	RPT event id used for service points call.
rptExecutable EntityProperties	RptExecutableEntity Properties	0..1	aggr	Describes the implemented code preparation for rapid prototyping at ExecutableEntity invocation. **Tags:** atp.Splitkey=rptExecutableEntityProperties atp.Status=obsolete
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the RptImplPolicy of a RptExecutableEvent for service based bypassing. **Tags:** atp.Splitkey=rptImplPolicy atp.Status=obsolete

▽

△

Class	**RptExecutableEntityEvent**			
rptServicePoint Post	RptServicePoint	*	ref	This describes the applicable Post Service Points for a RTEEvent / BswEvent of a bypassed ExecutableEntity. **Tags:** atp.Splitkey=rptServicePointPost
rptServicePoint Pre	RptServicePoint	*	ref	This describes the applicable Pre Service Points for a RTEEvent / BswEvent of a bypassed ExecutableEntity. **Tags:** atp.Splitkey=rptServicePointPre

Table A.859: RptExecutableEntityEvent

Class	**RptHook**			
Note	This meta-class provide the ability to describe a rapid prototyping hook. This can either be described by an other AUTOSAR system with the category RPT_SYSTEM or as a non AUTOSAR software.			
Base	*ARObject*			
Aggregated by	RptContainer.rptHook			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
codeLabel	CIdentifier	0..1	attr	This attribute provides a code label which is used in the implementation of the hook. For example this can be an C function name or the name of data definition.
mcdIdentifier	NameToken	0..1	attr	This attribute provides an identifier which shall be used in a MCD System to display the Rpt Hook.
rptArHook	AtpFeature	0..1	iref	This describes the hook with the means of another AUTOSAR system. **Tags:** atp.Splitkey=rptArHook, rptArHook.context Element, rptArHook.target **InstanceRef implemented by:** AnyInstanceRef
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. **Tags:** atp.Splitkey=sdg, sdg.sdgCaption.shortName

Table A.860: RptHook

Class	**RptServicePoint**			
Note	Description of a Service Point implemented for rapid prototyping.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RptSupportData.rptServicePoint			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
serviceId	PositiveInteger	0..1	attr	Unique ID (Range: 0 ... 65535) representing the service point.
symbol	CIdentifier	0..1	attr	Complete symbol of the function implementing the service point. This symbol is used for post-build hooking purposes.

Table A.861: RptServicePoint

Class	**RptSupportData**
Note	Root element for rapid prototyping support data related to one Implementation artifact on an ECU, in particular the RTE. The rapid prototyping support data may reference to elements provided for Mc SupportData.

▽

△

Class	**RptSupportData**			
Base	*ARObject*			
Aggregated by	McSupportData.rptSupportData			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
execution Context	RptExecutionContext	*	aggr	Defines an environment for the execution of Executable Entites. **Tags:** atp.Splitkey=executionContext, execution Context.shortName
rptComponent	RptComponent	*	aggr	Description of components for which rapid prototyping support is implemented. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptComponent, rptComponent.shortName, rptComponent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptServicePoint	RptServicePoint	*	aggr	This aggregation represents the collection of service points associated with the enclosing RptSuportData **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rptServicePoint, rptServicePoint.shortName, rptServicePoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.862: RptSupportData

Enumeration	**RteApiReturnValueProvisionEnum**
Note	This meta-class provides values to control how return values from RTE APIs are provided. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*AbstractAccessPoint*.returnValueProvision
Literal	***Description***
noReturnValue Provided	The RTE API shall not provide a return value. **Tags:** atp.EnumerationLiteralIndex=1
returnValue Provided	The RTE API shall provide a return value. **Tags:** atp.EnumerationLiteralIndex=0

Table A.863: RteApiReturnValueProvisionEnum

Class	**RtePluginProps**			
Note	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	FlatInstanceDescriptor.rtePluginProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
associated CrossSwCluster ComRtePlugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling cross Software Cluster communication. **Tags:** atp.Splitkey=associatedCrossSwClusterComRte Plugin

▽

△

Class	**RtePluginProps**			
associatedRte Plugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling local Software Cluster communication or communication in a non-cluster ECU. **Tags:** atp.Splitkey=associatedRtePlugin

Table A.864: RtePluginProps

Class	«atpMixed» **RuleArguments**			
Note	This represents the arguments for a rule-based value specification.			
Base	*ARObject*			
Aggregated by	RuleBasedValueSpecification.arguments			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
v	Numerical	0..1	attr	This represents a numerical value for the RuleBased ValueSpecification.
vf	Numerical	0..1	attr	This represents a numerical value for the RuleBased ValueSpecification which may subject to variability. The latest binding time of the VariationPoint shall be pre CompileTime. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
vt	VerbatimString	0..1	attr	This represents a textual value for the RuleBasedValue Specification.
vtf	NumericalOrText	0..1	aggr	This aggregation represents the ability to provide a value that is either numerical or text which existence is subject to variability. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.865: RuleArguments

Class	**RuleBasedAxisCont**			
Note	This represents the values for the axis of a compound primitive (curve, map). For standard and fix axes, SwAxisCont contains the values of the axis directly. The axis values of SwAxisCont with the category COM_AXIS, RES_AXIS are for display only. For editing and processing, only the values in the related GroupAxis are binding.			
Base	*ARObject*			
Aggregated by	ApplicationRuleBasedValueSpecification.swAxisCont			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	CalprmAxisCategory Enum	0..1	attr	This category specifies the particular axis types: • STD_AXIS • COM_AXIS • RES_AXIS (swArraysize necessary) **Tags:** xml.sequenceOffset=20

▽

△

Class	**RuleBasedAxisCont**			
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the axis of a compound primitive (curve, map). **Tags:** xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=80 xml.typeWrapperElement=false
swArraysize	ValueList	0..1	aggr	For multidimensional compound primitives (curve, map ...) it is necessary to know the dimensions.They are specified using swArraySize. **Tags:** xml.sequenceOffset=40
swAxisIndex	AxisIndexType	0..1	attr	This property allows to explicitly assign the axis contents to a particular axis. It is specified by numbers where 1 corresponds to the x-axis. It is also possible to derive the axis association from the sequence of the parent. **Tags:** xml.sequenceOffset=50
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:** xml.sequenceOffset=30

Table A.866: RuleBasedAxisCont

Class	**RuleBasedValueCont**			
Note	This represents the values of a compound primitive (CURVE, MAP, CUBOID, CUBE_4, CUBE_5, VAL_BLK) or an array.			
Base	*ARObject*			
Aggregated by	ApplicationRuleBasedValueSpecification.swValueCont			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the array or compound primitive (CURVE, MAP, CUBOID, CUBE_4, CUBE_5, VAL_BLK). **Tags:** atp.Splitkey=ruleBasedValues xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=80 xml.typeWrapperElement=false
swArraysize	ValueList	0..1	aggr	This attribute defines the size of each dimension for compound primitives CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK. For each dimension one value has to be defined, e.g. one in case of COM_AXIS and two or more in case of MAP. **Tags:** atp.Splitkey=swArraysize xml.sequenceOffset=40
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:** atp.Splitkey=unit xml.sequenceOffset=30

Table A.867: RuleBasedValueCont

Class	RuleBasedValueSpecification			
Note	This meta-class is used to support a rule-based initialization approach for data types with an array-nature (ApplicationArrayDataType and ImplementationDataType of category ARRAY) or a compound Application PrimitiveDataType (which also boils down to an array-nature).			
Base	*ARObject*			
Aggregated by	NumericalRuleBasedValueSpecification.ruleBasedValues, RuleBasedAxisCont.ruleBasedValues, RuleBasedValueCont.ruleBasedValues			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arguments	RuleArguments	0..1	aggr	This represents the arguments for the RuleBasedValue Specification. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arguments vh.latestBindingTime=preCompileTime xml.sequenceOffset=30
maxSizeToFill	Integer	0..1	attr	If a rule is chosen which does not fill until the end, this determines until which size the rule shall fill the values. **Tags:** xml.sequenceOffset=40
rule	Identifier	0..1	attr	This denotes the name of the rule of the RuleBasedValue Specification. The rule determines the calculation specification according which the arguments are used to calculated the values. **Tags:** xml.sequenceOffset=20

Table A.868: RuleBasedValueSpecification

Class	RunnableEntity			
Note	A `RunnableEntity` represents the smallest code-fragment that is provided by an `AtomicSwComponentType` and are executed under control of the RTE. `RunnableEntity`s 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*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.runnable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	RunnableEntityArgument	*	aggr	This represents the formal definition of an argument to a RunnableEntity.
asynchronousServerCallResultPoint	AsynchronousServerCallResultPoint	*	aggr	The server call result point admits a runnable to fetch the result of an asynchronous server call. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=asynchronousServerCallResultPoint, asynchronousServerCallResultPoint.shortName, asynchronousServerCallResultPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.



△

Class	**RunnableEntity**			
canBeInvoked Concurrently	Boolean	0..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 `AtomicSwComponentType`). This implies that it is the responsibility of the implementation of the `RunnableEntity` to take care of this form of concurrency.
dataRead Access	VariableAccess	*	aggr	RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReadAccess, dataReadAccess.shortName, dataReadAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointBy Argument	VariableAccess	*	aggr	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. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReceivePointByArgument, dataReceivePointByArgument.shortName, dataReceivePointByArgument.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointByValue	VariableAccess	*	aggr	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 the return value. The aggregation of dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReceivePointByValue, dataReceivePointByValue.shortName, dataReceivePointByValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
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:** atpVariation **Tags:** atp.Splitkey=dataSendPoint, dataSendPoint.shortName, dataSendPoint.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:** atpVariation **Tags:** atp.Splitkey=dataWriteAccess, dataWriteAccess.short Name, dataWriteAccess.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:** atpVariation **Tags:** atp.Splitkey=externalTriggeringPoint, externalTriggering Point.ident.shortName, externalTriggeringPoint.variation Point.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:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint, internalTriggering Point.shortName, internalTriggeringPoint.variation Point.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:** atpVariation **Tags:** atp.Splitkey=modeAccessPoint, modeAccess Point.ident.shortName, modeAccessPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
modeSwitch Point	ModeSwitchPoint	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeSwitchPoint, modeSwitchPoint.shortName, modeSwitchPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
parameter Access	ParameterAccess	*	aggr	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. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=parameterAccess, parameterAccess.shortName, parameterAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
readLocal Variable	VariableAccess	*	aggr	The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=readLocalVariable, readLocalVariable.shortName, readLocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
serverCallPoint	ServerCallPoint	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serverCallPoint, serverCallPoint.shortName, serverCallPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
symbol	CIdentifier	0..1	attr	The symbol describing this `RunnableEntity`'s entry point. This is considered the API of the `RunnableEntity` and is required during the RTE contract phase.
waitPoint	WaitPoint	*	aggr	The `WaitPoint` associated with the `RunnableEntity`. **Tags:** atp.Splitkey=waitPoint, waitPoint.shortName

▽

△

Class	**RunnableEntity**			
writtenLocal Variable	VariableAccess	*	aggr	The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=writtenLocalVariable, writtenLocal Variable.shortName, writtenLocalVariable.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.869: RunnableEntity

Class	**RunnableEntityArgument**			
Note	This meta-class represents the ability to provide specific information regarding the arguments to a RunnableEntity.			
Base	*ARObject*			
Aggregated by	RunnableEntity.argument			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
direction	ArgumentDirection Enum	0..1	attr	This attribute denotes the direction of the argument.
implementation DataType	ImplementationData Type	0..1	ref	This reference represents the implementationDataType used to type the RunnableEntityArgument if the Runnable EntityArgument represents an argument of a ClientServer Operation, rather than a port-defined argument value.
serverArgument ImplPolicy	ServerArgumentImpl PolicyEnum	0..1	attr	This attribute defines how the argument type of the server's RunnableEntity is implemented. 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.
symbol	CIdentifier	0..1	attr	This represents the symbol to be generated into the actual signature on the level of the C programming language.

Table A.870: RunnableEntityArgument

Enumeration	**RxAcceptContainedIPduEnum**
Note	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
Aggregated by	ContainerIPdu.rxAcceptContainedIPdu
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

▽

△

Enumeration	**RxAcceptContainedIPduEnum**
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 A.871: RxAcceptContainedIPduEnum

Class	**RxIdentifierRange**			
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	*ARObject*			
Aggregated by	CanFrameTriggering.rxIdentifierRange, CanXlNmNodeProps.rxIdentifierRange, IEEE1722TpAcfCanPart.canIdentifierRange			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
lowerCanId	PositiveInteger	0..1	attr	This attribute can be used together with the upperCanId attribute to define a range of CanIds.
upperCanId	PositiveInteger	0..1	attr	This attribute can be used together with the lowerCanId attribute to define a range of CanIds.

Table A.872: RxIdentifierRange

Class	**SOMEIPTransformationDescription**			
Note	The SOMEIPTransformationDescription is used to specify SOME/IP transformer specific attributes.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
alignment	PositiveInteger	0..1	attr	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. The alignment shall be specified in Bits.
byteOrder	ByteOrderEnum	0..1	attr	Defines which byte order shall be serialized by the SOME/IP transformer
interfaceVersion	PositiveInteger	0..1	attr	The interface version the SOME/IP transformer shall use.

Table A.873: SOMEIPTransformationDescription

Class	«atpVariation» **SOMEIPTransformationISignalProps**			
Note	The class SOMEIPTransformationISignalProps specifies ISignal specific configuration properties for SOME/IP transformer attributes.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
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.

▽

△

Class	«atpVariation» **SOMEIPTransformationISignalProps**			
sizeOfArray LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of fixed-size arrays or dynamic size arrays in the SOME/IP message. This attribute is valid for all available occurrences of fixed-size arrays or dynamic size arrays in the SOME/IP message.
sizeOfString LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of dynamic or fixed length strings in the SOME/IP message. This attribute is valid for all available occurrences of strings in the SOME/IP message.
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 Definition	TlvDataIdDefinitionSet	*	ref	This reference identifies the TlvDataIdDefinitions relevant for the enclosing SOMEIPTransformationISignalProps **Tags:** atp.Splitkey=tlvDataIdDefinition

Table A.874: SOMEIPTransformationISignalProps

Class	**SOMEIPTransformationProps**			
Note	The class SOMEIPTransformationProps specifies SOME/IP specific configuration properties.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TransformationProps*			
Aggregated by	TransformationPropsSet.transformationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
alignment	PositiveInteger	0..1	attr	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. The alignment shall be specified in Bits.
sizeOfArray LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced Array in the SOME/IP message.
sizeOfString LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced String in the SOME/IP message.
sizeOfStruct LengthField	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.
sizeOfUnion LengthField	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 A.875: SOMEIPTransformationProps

Class	**ScheduleTableEntry** (abstract)			
Note	Table entry in a LinScheduleTable. Specifies what will be done in the frame slot. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	ApplicationEntry, *FreeFormatEntry*, *LinConfigurationEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
delay	TimeValue	0..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:** atp.Splitkey=introduction xml.sequenceOffset=-10
positionInTable	Integer	0..1	attr	Relative position in the schedule table. The first entry index in the schedule table is 0.

Table A.876: ScheduleTableEntry

Class	**SdgClass**			
Note	An SdgClass specifies the name and structure of the SDG that may be used to store proprietary data in an AUTOSAR model. The SdgClass is similar to an UML stereotype.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SdgElementWithGid*			
Aggregated by	SdgDef.sdgClass			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
attribute (ordered)	SdgAttribute	*	aggr	Defintion of the structure of the Sdg **Tags:** xml.sequenceOffset=30
caption	Boolean	0..1	attr	Specifies if a caption is required. Note: only Sdgs that have a caption can be referenced **Tags:** xml.sequenceOffset=20
extendsMeta Class	MetaClassName	0..1	attr	The AUTOSAR Meta-Class that may be extended by this SdgClass. **Tags:** xml.sequenceOffset=10
sdgConstraint	TraceableText	*	ref	Semantic constraints that restrict the structure of the special data group. **Tags:** atp.Splitkey=sdgConstraint xml.sequenceOffset=40

Table A.877: SdgClass

Class	**SdgDef**			
Note	A SdgDef groups several SdgClasses which belong to the same extension. The concept of an SdgDef is similiar to an UML Profile. **Tags:** atp.recommendedPackage=SdgDefs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**SdgDef**			
sdgClass	SdgClass	*	aggr	The owned sdgClasses which define the structure of the Sdgs **Tags:** atp.Splitkey=sdgClass, sdgClass.shortName xml.namePlural=SDG-CLASSES

Table A.878: SdgDef

Primitive	**SectionInitializationPolicyType**
Note	SectionInitializationPolicyType describes the intended initialization of MemorySections. The following values are standardized in AUTOSAR Methodology: • **INIT**: To be used for (explicitly or not explicitly) initialized variables. • **CLEARED**: To be used for not explicitly initialized variables. • **POWER-ON-CLEARED**: To be used for variables that are not explicitly initialized (cleared) during normal start-up. Instead these are cleared only after power on reset. Please note that the values are defined similar to the representation of enumeration types in the XML schema to ensure backward compatibility. **Tags:** xml.xsd.customType=SECTION-INITIALIZATION-POLICY-TYPE xml.xsd.type=NMTOKEN

Table A.879: SectionInitializationPolicyType

Class	**SectionNamePrefix**			
Note	A prefix to be used for generated code artifacts defining a memory section name in the source code of the using module or SWC.			
Base	*ARObject*, *ImplementationProps*, *Referrable*			
Aggregated by	ResourceConsumption.sectionNamePrefix			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
implementedIn	DependencyOnArtifact	0..1	ref	Optional reference that allows to Indicate the code artifact (header file) containing the preprocessor implementation of memory sections with this prefix. The usage of this link supersedes the usage of a memory mapping header with the default name (derived from the BswModuleDescription's shortName). **Tags:** atp.Splitkey=implementedIn

Table A.880: SectionNamePrefix

Class	**SecureCommunicationAuthenticationProps**			
Note	Authentication properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.authenticationProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
authInfoTx Length	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 A.881: SecureCommunicationAuthenticationProps

Class	**SecureCommunicationFreshnessProps**			
Note	Freshness properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.freshnessProps			
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 A.882: SecureCommunicationFreshnessProps

Class	**SecureCommunicationProps**			
Note	This meta-class contains configuration settings that are specific for an individual SecuredIPdu.			
Base	*ARObject*			
Aggregated by	SecuredIPdu.secureCommunicationProps			
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 counting is done according to TPS_SYST_01068.
authentication BuildAttempts	PositiveInteger	0..1	attr	This attribute specifies the number of authentication build attempts.
authentication Retries	PositiveInteger	0..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	0..1	attr	This attribute defines a numerical identifier for the Secured I-PDU.

▽

△

Class	**SecureCommunicationProps**			
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.
use Authenticated Encryption	Boolean	0..1	attr	A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If it set to TRUE, Authenticated Encryption with Associated Data (AEAD) shall be applied.

Table A.883: SecureCommunicationProps

Class	**SecuredIPdu**			
Note	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. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Props	SecureCommunication AuthenticationProps	0..1	ref	Reference to authentication properties that are valid for this SecuredIPdu. **Tags:** atp.Splitkey=authenticationProps
dynamic RuntimeLength Handling	Boolean	0..1	attr	Defines whether the length information for handling this SecuredIPdu with SecuredIPdu.useSecuredPdu Header=noHeader is taken from the configuration or from the actually provided length information during runtime. true: SecuredIPdu length information is taken from the actually provided length information during runtime. false: SecuredIPdu length information is taken from the configuration.
freshnessProps	SecureCommunication FreshnessProps	0..1	ref	Reference to freshness properties that are valid for this SecuredIPdu. **Tags:** atp.Splitkey=freshnessProps

▽

△

Class	**SecuredIPdu**			
payload	PduTriggering	0..1	ref	Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks. **Tags:** atp.Splitkey=payload
secure Communication Props	SecureCommunication Props	0..1	aggr	Specific configuration properties for this SecuredIPdu. **Tags:** atp.Splitkey=secureCommunicationProps
useAs Cryptographic IPdu	Boolean	0..1	attr	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.
useSecuredPdu Header	SecuredPduHeader Enum	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 A.884: SecuredIPdu

Class	**SecurityEventContextProps**			
Note	This meta-class specifies the SecurityEventDefinition to be mapped to an IdsmInstance and adds mapping-dependent properties of this security event valid only for this specific mapping.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*SecurityEventContextMapping*.mappedSecurityEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
default ReportingMode	SecurityEventReporting ModeEnum	0..1	attr	This attribute defines the default reporting mode for the referenced security event.
persistent Storage	Boolean	0..1	attr	This attribute controls whether qualified reportings of the referenced security event shall be stored persistently by the mapped IdsmInstance or not.
securityEvent	SecurityEventDefinition	0..1	ref	This reference defines the security event that is mapped and enriched by SecurityEventMappingProps with mapping dependent properties. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEvent, securityEvent.securityEvent Definition, securityEvent.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
sensorInstance Id	PositiveInteger	0..1	attr	This attribute defines the ID of the security sensor that detects the referenced security event.
severity	PositiveInteger	0..1	attr	This attribute defines how critical/severe the referenced security event is. Please note that currently, the severity level meanings of specific integer values is not specified by AUTOSAR but left to the party responsible for the IDS system design (e.g. the OEM).

Table A.885: SecurityEventContextProps

Class	SegmentPosition			
Note	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.			
Base	*ARObject*			
Aggregated by	*MultiplexedPart*.segmentPosition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
segmentByte Order	ByteOrderEnum	0..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	0..1	attr	Data Length of the segment in bits.
segment Position	Integer	0..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 A.886: SegmentPosition

Class	***SenderComSpec*** (abstract)			
Note	Communication attributes for a sender port (`PPortPrototype` typed by `SenderReceiverInterface`).			
Base	*ARObject*, *PPortComSpec*			
Subclasses	NonqueuedSenderComSpec, QueuedSenderComSpec			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a SenderComSpec. **Tags:** atp.Splitkey=compositeNetworkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
dataElement	AutosarDataPrototype	0..1	ref	Data element these quality of service attributes apply to. **Stereotypes:** atpIdentityContributor
handleOutOf Range	HandleOutOfRange Enum	0..1	attr	This attribute controls how out-of-range values shall be dealt with. This Attribute is only used by the AUTOSAR Classic Platform.
network Representation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus. **Tags:** atp.Splitkey=networkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	***SenderComSpec*** (abstract)			
transmission Acknowledge	Transmission Acknowledgement Request	0..1	aggr	Requested transmission acknowledgement for data element. **Tags:** atp.Splitkey=transmissionAcknowledge This Attribute is only used by the AUTOSAR Classic Platform.
transmission Props	TransmissionComSpec Props	0..1	aggr	This aggregation represents the definition transmission props in the context of the enclosing SenderComSpec. **Tags:** atp.Splitkey=transmissionProps
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 A.887: SenderComSpec

Class	**SenderRecArrayElementMapping**			
Note	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 Sender ReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference to the Application ArrayElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference to the ImplementationArrayElement shall be used. 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. Regardless whether composite or primitive array element is mapped the indexed element always needs to be specified.			
Base	*ARObject*			
Aggregated by	SenderRecArrayTypeMapping.arrayElementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
complexType Mapping	SenderRecComposite TypeMapping	0..1	aggr	This aggregation will be used if the element is composite. **Tags:** atp.Splitkey=complexTypeMapping
indexedArray Element	IndexedArrayElement	0..1	aggr	Reference to an indexed array element in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=indexedArrayElement
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationArrayElement. **Tags:** atp.Splitkey=systemSignal

Table A.888: SenderRecArrayElementMapping

Class	***SenderRecCompositeTypeMapping*** (abstract)
Note	Two mappings exist for the composite data types: "ArrayTypeMapping" and "RecordTypeMapping". In both, a primitive datatype will be mapped to a system signal. 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. 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 CompositeType Mapping or aggregation between the RecordElementMapping and CompositeTypeMapping).
Base	*ARObject*

▽

△

Class	***SenderRecCompositeTypeMapping*** (abstract)			
Subclasses	SenderRecArrayTypeMapping, SenderRecRecordTypeMapping			
Aggregated by	SenderRecArrayElementMapping.complexTypeMapping, SenderReceiverToSignalGroupMapping.typeMapping, SenderRecRecordElementMapping.complexTypeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.889: SenderRecCompositeTypeMapping

Class	**SenderRecRecordElementMapping**			
Note	Mapping of a primitive record element to a SystemSignal. If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference applicationRecordElement 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. 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.			
Base	*ARObject*			
Aggregated by	SenderRecRecordTypeMapping.recordElementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationRecordElement	ApplicationRecordElement	0..1	ref	Reference to an ApplicationRecordElement in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=applicationRecordElement
complexTypeMapping	SenderRecCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite. **Tags:** atp.Splitkey=complexTypeMapping
implementationRecordElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationRecordElement in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=implementationRecordElement
senderToSignalTextTableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal. **Tags:** atp.Splitkey=senderToSignalTextTableMapping
signalToReceiverTextTableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype. **Tags:** atp.Splitkey=signalToReceiverTextTableMapping
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement. **Tags:** atp.Splitkey=systemSignal

Table A.890: SenderRecRecordElementMapping

Class	***SenderReceiverAnnotation*** (abstract)
Note	Annotation of the data elements in a port that realizes a sender/receiver interface.
Base	*ARObject*, *GeneralAnnotation*
Subclasses	ReceiverAnnotation, SenderAnnotation

▽

△

Class	***SenderReceiverAnnotation*** (abstract)			
Aggregated by	*PortPrototype*.senderReceiverAnnotation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
computed	Boolean	0..1	attr	Flag whether this data element was not measured directly but instead was calculated from possibly several other measured or calculated values.
dataElement	VariableDataPrototype	0..1	ref	The instance of VariableDataPrototype annotated. **Tags:** atp.Splitkey=dataElement
limitKind	DataLimitKindEnum	0..1	attr	This min or max has not to be mismatched with the min- and max for data-value in a compu-method. For example, this annotation shows when the result of the calculation performed in a RunnableEntity owned by one AtomicSw ComponentType is transmitted to another AtomicSw ComponentType whose RunnableEntity will use this value as a limit, e.g. the max.power which can be used by that software-component, or the current min. slip.
processingKind	ProcessingKindEnum	0..1	attr	This attribute controls how data is processed according to the possible values of ProcessingKindEnum.

Table A.891: SenderReceiverAnnotation

Class	**SenderReceiverInterface**			
Note	A sender/receiver interface declares a number of data elements to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	aggr	The data elements of this `SenderReceiverInterface`. **Tags:** atp.Splitkey=dataElement, dataElement.shortName
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement **Tags:** atp.Splitkey=invalidationPolicy
metaDataItemSet	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with `dataElement`s of the enclosing `SenderReceiverInterface` **Tags:** atp.Splitkey=metaDataItemSet

Table A.892: SenderReceiverInterface

Class	**SenderReceiverToSignalGroupMapping**			
Note	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	SenderReceiverToSignalGroupMapping			
dataElement	VariableDataPrototype	0..1	iref	Reference to a data element with a composite datatype which is mapped to a signal group. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
signalGroup	SystemSignalGroup	0..1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type **Tags:** atp.Splitkey=signalGroup
typeMapping	SenderRecComposite TypeMapping	0..1	aggr	The CompositeTypeMapping maps the ApplicationArray Elements and ApplicationRecordElements to Signals of the SignalGroup. **Tags:** atp.Splitkey=typeMapping

Table A.893: SenderReceiverToSignalGroupMapping

Class	SenderReceiverToSignalMapping			
Note	Mapping of a sender receiver communication data element to a signal.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	VariableDataPrototype	0..1	iref	Reference to the data element. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal. **Tags:** atp.Splitkey=senderToSignalTextTableMapping
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype. **Tags:** atp.Splitkey=signalToReceiverTextTableMapping
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the data element. **Tags:** atp.Splitkey=systemSignal

Table A.894: SenderReceiverToSignalMapping

Class	SensorActuatorSwComponentType
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*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*
Aggregated by	ARPackage.element

▽

△

Class	**SensorActuatorSwComponentType**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
sensorActuator	HwDescriptionEntity	0..1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware. **Tags:** atp.Splitkey=sensorActuator

Table A.895: SensorActuatorSwComponentType

Enumeration	**ServerArgumentImplPolicyEnum**
Note	This defines how the argument type of the server's `RunnableEntity` is implemented.
Aggregated by	ArgumentDataPrototype.serverArgumentImplPolicy, RunnableEntityArgument.serverArgumentImplPolicy
Literal	***Description***
useArgumentType	The argument type of the `RunnableEntity` is derived from the `AutosarDataType` of the `ArgumentDataPrototype`. **Tags:** atp.EnumerationLiteralIndex=0
useVoid	The argument type of the `RunnableEntity` is void. **Tags:** atp.EnumerationLiteralIndex=2

Table A.896: ServerArgumentImplPolicyEnum

Class	***ServerCallPoint*** (abstract)			
Note	If a RunnableEntity owns a `ServerCallPoint` it is entitled to invoke a particular `ClientServerOperation` of a specific `RPortPrototype` of the corresponding `AtomicSwComponentType`. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AsynchronousServerCallPoint, SynchronousServerCallPoint			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	iref	The operation that is called by this runnable. **Tags:** atp.Splitkey=operation, operation.contextRPort, operation.targetRequiredOperation **InstanceRef implemented by:** ROperationInAtomicSwcInstanceRef
timeout	TimeValue	0..1	attr	Time in seconds before the server call times out and returns with an error message. It depends on the call type (synchronous or asynchronous) how this is reported.

Table A.897: ServerCallPoint

Class	**ServerComSpec**			
Note	Communication attributes for a server port (`PPortPrototype` and `ClientServerInterface`).			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ServerComSpec**			
operation	ClientServerOperation	0..1	ref	Operation these communication attributes apply to. **Stereotypes:** atpIdentityContributor
queueLength	PositiveInteger	0..1	attr	Length of call queue on the server side. The queue is implemented by the RTE. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation. **Tags:** atp.Splitkey=transformationComSpecProps

Table A.898: ServerComSpec

Class	***ServiceDependency*** (abstract)			
Note	Collects all dependencies of a software module or component on an AUTOSAR Service related to a specific item (e.g. an NVRAM Block, a diagnostic event etc.). It defines the quality of service (Service Needs) of this item as well as (optionally) references to additional elements. This information is required for tools in order to generate the related basic software configuration and ServiceSwComponentTypes.			
Base	*ARObject*			
Subclasses	BswServiceDependency, SwcServiceDependency			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignedData Type	RoleBasedDataType Assignment	0..1	aggr	This is the role of the assignment data type in the given context. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedDataType, assignedData Type.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
diagnostic Relevance	ServiceDiagnostic RelevanceEnum	0..1	attr	If this attribute indicates a relevance for diagnostics then the integrator has a much easier time identifying the candidates for the configuration of the diagnostic stack. Example: identification of mode conditions (e.g. communication between application and BswM) relevant for the Dcm.
symbolicName Props	SymbolicNameProps	0..1	aggr	This attribute can be taken to contribute to the creation of symbolic name values. **Tags:** atp.Splitkey=symbolicNameProps, symbolicName Props.shortName

Table A.899: ServiceDependency

Class	***ServiceNeeds*** (abstract)
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*

▽

△

Class	**ServiceNeeds** (abstract)			
Subclasses	BswMgrNeeds, ChargeManagerNeeds, ComMgrUserNeeds, CryptoKeyManagementNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, *DiagnosticCapabilityElement*, DltUserNeeds, *DoIpServiceNeeds*, EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, GeneralPurposeTimerServiceNeeds, GlobalSupervisionNeeds, IdsMgrCustomTimestampNeeds, IdsMgrNeeds, IndicatorStatusNeeds, J1939DcmDm19Support, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequestServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpointNeeds, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, VendorSpecificServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.900: ServiceNeeds

Class	**ServiceProxySwComponentType**			
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 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. Apart from the semantics, a ServiceProxySwComponentType has these specific properties: • 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. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.901: ServiceProxySwComponentType

Class	**ServiceSwComponentType**			
Note	ServiceSwComponentType is used for configuring services for a given ECU. Instances of this class are only to be created in ECU Configuration phase for the specific purpose of the service configuration. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.902: ServiceSwComponentType

Enumeration	**ServiceVersionAcceptanceKindEnum**
Note	Defined the possible acceptance kinds for required service instances.
Aggregated by	ConsumedServiceInstance.versionDrivenFindBehavior, RequiredSomeipServiceInstance.version DrivenFindBehavior
Literal	***Description***
exactOrAnyMinor Version	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. **Tags:** atp.EnumerationLiteralIndex=0
minimumMinor Version	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. **Tags:** atp.EnumerationLiteralIndex=1

Table A.903: ServiceVersionAcceptanceKindEnum

Class	**ShortNameFragment**			
Note	This class describes how the Referrable.shortName is composed of several shortNameFragments.			
Base	*ARObject*			
Aggregated by	*Referrable*.shortNameFragment			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
fragment	Identifier	0..1	attr	This specifies a single shortName (fragment) which is part of the composed shortName. **Tags:** xml.sequenceOffset=20
role	String	0..1	attr	This specifies the role of fragment to define e.g. the order of the fragments. **Tags:** xml.sequenceOffset=10

Table A.904: ShortNameFragment

Enumeration	**SignalServiceTranslationControlEnum**
Note	This enumeration allows to define how the service instance offer/subscribe control shall behave.
Aggregated by	SignalServiceTranslationProps.serviceControl
Literal	***Description***
allPartialNetworks Active	Defines the start of service control when all specified partial networks are active. **Tags:** atp.EnumerationLiteralIndex=3 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
anyPartialNetwork Active	Defines the start of service control when any specified partial network is active. **Tags:** atp.EnumerationLiteralIndex=4 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
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 A.905: SignalServiceTranslationControlEnum

Class	SignalServiceTranslationElementProps			
Note	Defined translation properties for individual mapped elements. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationEventProps.elementProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
element	DataPrototype Reference	0..1	aggr	Reference to the leaf element the SignalService TranslationElementProps apply to. **Tags:** atp.Splitkey=element
filter	DataFilter	0..1	aggr	Defines an optional filter to be applied during translation. **Tags:** atp.Splitkey=filter
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 A.906: SignalServiceTranslationElementProps

Class	SignalServiceTranslationEventProps			
Note	This element allows to define the properties which are applicable for the signal/service translation event.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationProps.signalServiceTranslationEventProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
elementProps	SignalService TranslationElement Props	*	aggr	Defines properties for a single translated element. **Tags:** atp.Splitkey=elementProps, elementProps.short Name This Attribute is only used by the AUTOSAR Classic Platform.
safeTranslation	Boolean	0..1	attr	Defined whether the translation shall happen in a safe way.
secure Translation	Boolean	0..1	attr	Defined whether the translation shall happen in a secure way.
translation Target	VariableDataPrototype	0..1	iref	Reference to a VariableDataPrototype representing the target of signal/service translation. **Tags:** atp.Splitkey=translationTarget, translation Target.contextComponent, translationTarget.context Composition, translationTarget.contextPort, translation Target.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table A.907: SignalServiceTranslationEventProps

Class	SignalServiceTranslationProps			
Note	This element allows to define the properties which are applicable for the signal/service translation service.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationPropsSet.signalServiceTranslationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

△

Class	**SignalServiceTranslationProps**			
control Consumed EventGroup	ConsumedEventGroup	*	ref	Reference to the EventGroup which encapsulates the signal-based payload. **Tags:** atp.Splitkey=controlConsumedEventGroup This Attribute is only used by the AUTOSAR Classic Platform.
controlPnc	PncMappingIdent	*	ref	Reference to the PNCs which control the offer/subscribe behavior of the translated service instance. **Tags:** atp.Splitkey=controlPnc This Attribute is only used by the AUTOSAR Classic Platform.
controlProvided EventGroup	EventHandler	*	ref	Reference to the provided event group (aka Event Handler) which is automatically available when service Control equals translationStart. **Tags:** atp.Splitkey=controlProvidedEventGroup This Attribute is only used by the AUTOSAR Classic Platform.
serviceControl	SignalService TranslationControlEnum	0..1	attr	Defines how the service instance control shall behave.
signalService Translation EventProps	SignalService TranslationEventProps	*	aggr	Defines properties for a single translated event. **Tags:** atp.Splitkey=signalServiceTranslationEventProps, signalServiceTranslationEventProps.shortName

Table A.908: SignalServiceTranslationProps

Class	**SimulatedExecutionTime**			
Note	Specifies the ExecutionTime which has been gathered using simulation means.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
maximum ExecutionTime	MultidimensionalTime	0..1	aggr	The maximum simulated execution time. **Tags:** atp.Splitkey=maximumExecutionTime
minimum ExecutionTime	MultidimensionalTime	0..1	aggr	The minimum simulated execution time. **Tags:** atp.Splitkey=minimumExecutionTime
nominal ExecutionTime	MultidimensionalTime	0..1	aggr	The nominal simulated execution time. **Tags:** atp.Splitkey=nominalExecutionTime

Table A.909: SimulatedExecutionTime

Class	**SoAdConfig**			
Note	SoAd Configuration for one specific Physical Channel.			
Base	*ARObject*			
Aggregated by	EthernetPhysicalChannel.soAdConfig			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	SoAdConfig			
socketAddress	SocketAddress	*	aggr	Collection of SoAdAddresses. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=socketAddress, socketAddress.shortName, socketAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.910: SoAdConfig

Class	SoConIPduIdentifier			
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*			
Aggregated by	SocketConnectionIpduIdentifierSet.iPduIdentifier			
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. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.
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. **Tags:** atp.Splitkey=pduTriggering

Table A.911: SoConIPduIdentifier

Class	SocketAddress			
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 ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SoAdConfig.socketAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allowedIPv6Ext Headers	IPv6ExtHeaderFilterList	0..1	ref	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. **Tags:** atp.Splitkey=allowedIPv6ExtHeaders

▽

△

Class	**SocketAddress**			
allowedTcp Options	TcpOptionFilterList	0..1	ref	Reference to a list of TCP options allowed for this Socket Connection. **Tags:** atp.Splitkey=allowedTcpOptions This Attribute is only used by the AUTOSAR Classic Platform.
application Endpoint	ApplicationEndpoint	0..1	aggr	Application addressing **Tags:** atp.Splitkey=applicationEndpoint, application Endpoint.shortName
connector	EthernetCommunication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP unicast address for an ECU that is part of the model. **Tags:** atp.Splitkey=connector
differentiated ServiceField	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.
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.
multicast Connector	EthernetCommunication Connector	*	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP multicast address, i.e. if the aggregated ApplicationEndpoint references a NetworkEndpoint that describes an IP Address in the IP multicast range. Such a SocketAddress contains references to those Ecus (via the multicastConnector reference) in the model that will receive multicast messages via the SocketAddress that is defined by the aggregated ApplicationEndpoint and NetworkEndpoint, i.e. IP Address and UDP Port combination. **Tags:** atp.Splitkey=multicastConnector
pduCollection MaxBufferSize	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.
pduCollection Timeout	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.
staticSocket Connection	StaticSocketConnection	*	aggr	Definition of a static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticSocketConnection, staticSocket Connection.shortName, staticSocketConnection.variation Point.shortLabel vh.latestBindingTime=postBuild
ttl	PositiveInteger	0..1	attr	This attribute defines a value set in the header of an Internet Protocol (IP) packet that tells network devices the maximum number of router hops the packet can make before it is discarded. The TTL value is a counter that is decremented by 1 every time the packet passes through a router.

▽

△

Class	**SocketAddress**			
udpChecksum Handling	UdpChecksum CalculationEnum	0..1	attr	Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksum Disabled) on the related socket connection.

Table A.912: SocketAddress

Class	**SoftwareContext**			
Note	Specifies the context of the software for this resource consumption.			
Base	*ARObject*			
Aggregated by	*ExecutionTime*.softwareContext, *HeapUsage*.softwareContext, *StackUsage*.softwareContext			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
input	String	0..1	attr	Specifies the input vector which is used to provide the ExecutionTime.
state	String	0..1	attr	Specifies the state the software is in when the Execution Time is provided.

Table A.913: SoftwareContext

Class	**SomeipSdClientEventGroupTimingConfig**			
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.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
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). **Tags:** atp.Splitkey=requestResponseDelay
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 along as the Eventgroup is requested and no SubscribeEventGroupAck was received.
timeToLive	PositiveInteger	0..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 A.914: SomeipSdClientEventGroupTimingConfig

Class	**SomeipSdClientServiceInstanceConfig**
Note	Client specific settings that are relevant for the configuration of SOME/IP Service-Discovery. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*

▽

△

Class	**SomeipSdClientServiceInstanceConfig**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialFind Behavior	InitialSdDelayConfig	0..1	aggr	Controls initial find behavior of clients. **Tags:** atp.Splitkey=initialFindBehavior
priority	PositiveInteger	0..1	attr	This attribute defines the VLAN frame priority for Service Discovery messages that result from RequiredSomeip ServiceInstances that are referncing this SomeipSdClient ServiceInstanceConfig (Find, SubscribeEventGroup, Stop SubscribeEventgroup). Values from 0 (best effort) to 7 (highest) are allowed.
serviceFind TimeToLive	PositiveInteger	0..1	attr	This attribute represents the ability to define the time in seconds the service find is valid. Note! The TTL value for FindService entries is not used and shall be ignored by the server service. This configuration is only kept for backward compatibility. Default value if not specified shall be 0xFFFFFF. This Attribute is only used by the AUTOSAR Classic Platform.

Table A.915: SomeipSdClientServiceInstanceConfig

Class	**SomeipSdServerEventGroupTimingConfig**			
Note	EventGroup specific timing configuration settings. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
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). **Tags:** atp.Splitkey=requestResponseDelay

Table A.916: SomeipSdServerEventGroupTimingConfig

Class	**SomeipSdServerServiceInstanceConfig**			
Note	Server specific settings that are relevant for the configuration of SOME/IP Service-Discovery. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialOffer Behavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server. **Tags:** atp.Splitkey=initialOfferBehavior
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds) and greater then 0.

▽

△

Class	**SomeipSdServerServiceInstanceConfig**			
priority	PositiveInteger	0..1	attr	This attribute defines the VLAN frame priority for Service Discovery messages that result from ProvidedSomeip ServiceInstances that are referencing the SomeipSd ServerServiceInstanceConfig (OfferService, StopOffer Service, SubscribeEventGroupAck). Values from 0 (best effort) to 7 (highest) are allowed.
request ResponseDelay	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.Splitkey=requestResponseDelay
serviceOffer TimeToLive	PositiveInteger	0..1	attr	Defines the time in seconds the service offer is valid.

Table A.917: SomeipSdServerServiceInstanceConfig

Class	**SomeipServiceVersion**			
Note	This meta-class represents the ability to describe a version of a SOME/IP Service.			
Base	*ARObject*			
Aggregated by	ConsumedServiceInstance.blocklistedVersion, RequiredSomeipServiceInstance.blocklistedVersion, SomeipServiceInterfaceDeployment.serviceInterfaceVersion			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. **Tags:** xml.sequenceOffset=10
minorVersion	PositiveInteger	0..1	attr	Minor Version of the ServiceInterface. **Tags:** xml.sequenceOffset=20

Table A.918: SomeipServiceVersion

Class	**SomeipTpConnection**			
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*			
Aggregated by	SomeipTpConfig.tpConnection			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tpChannel	SomeipTpChannel	0..1	ref	Assignment of configuration properties valid for this SomeipTpConnection. **Tags:** atp.Splitkey=tpChannel
tpConcurrent ProcessingSdu	PduTriggering	*	ref	Reference to a set of IPdus that are used to implement concurrent processing of several upper layer IPdus related to one transportPdu by the SOME/IP TP segmenter. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConcurrentProcessingSdu, tpConcurrent ProcessingSdu.pduTriggering, tpConcurrentProcessing Sdu.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	SomeipTpConnection			
tpSdu	PduTriggering	0..1	ref	Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tp ConcurrentProcessingSdu references shall be used. **Tags:** atp.Splitkey=tpSdu
transportPdu	PduTriggering	0..1	ref	Reference to the segmented IPdu. **Tags:** atp.Splitkey=transportPdu

Table A.919: SomeipTpConnection

Class	SporadicEventTriggering			
Note	Describes the behavior of an event which occurs occasionally or singularly.			
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Timing Constraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
jitter	MultidimensionalTime	0..1	aggr	The maximum devation of the sporadic event occurrence. Jitter=max |nthPeriod - standardPeriod| **Tags:** atp.Splitkey=jitter xml.sequenceOffset=30
maximumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The maximum time distance between two consecutive (subsequent) occurrences of the associated event. **Tags:** atp.Splitkey=maximumInterArrivalTime xml.sequenceOffset=20
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The minimum time distance between two consecutive (subsequent) occurrences of the associated event. **Tags:** atp.Splitkey=minimumInterArrivalTime xml.sequenceOffset=10
period	MultidimensionalTime	0..1	aggr	The periodic distance between subsequent occurrences of the event. **Tags:** atp.Splitkey=period xml.sequenceOffset=40

Table A.920: SporadicEventTriggering

Class	StateDependentFirewall			
Note	Firewall rules that are defined in a firewall state **Tags:** atp.recommendedPackage=StateDependentFirewallRules			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defaultAction	FirewallActionEnum	0..1	attr	This attribute defines a defaultAction in case that the VehicleMode is not yet set.
firewallRule Props (ordered)	FirewallRuleProps	*	aggr	Collection of firewall rules that apply in the vehicle mode

▽

△

Class	**StateDependentFirewall**			
firewallState Mode Declaration	ModeDeclaration	*	ref	Reference to firewall states in which the Firewall is active. If one of the referenced ModeDeclarations is the current firewall state then the firewall rule shall be considered as active. **Tags:** atp.Splitkey=firewallStateModeDeclaration This Attribute is only used by the AUTOSAR Classic Platform.

Table A.921: StateDependentFirewall

Class	**StaticPart**			
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*			
Aggregated by	MultiplexedIPdu.staticPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPdu	ISignalIPdu	0..1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu. **Tags:** atp.Splitkey=iPdu

Table A.922: StaticPart

Class	**StaticSocketConnection**			
Note	Definition of static SocketConnection between the Socket that is defined by the aggregating Socket Address and the remoteAddress.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SocketAddress.staticSocketConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduIdentifier	SoConIPduIdentifier	*	ref	Assignment of IPduIdentifiers that are transmitted over the static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduIdentifier, iPduIdentifier.soConIPdu Identifier, iPduIdentifier.variationPoint.shortLabel vh.latestBindingTime=postBuild
remoteAddress	SocketAddress	0..1	ref	RemoteAddress of the static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=remoteAddress, remoteAddress.socket Address, remoteAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
tcpConnect Timeout	TimeValue	0..1	attr	Specifies the time in seconds how long TCP connect attempts are repeated to reach SOAD_SOCON_ONLINE. This attribute is restricted to socket connection groups which are initiating a TCP connection and are under control of SoAd.
tcpRole	TcpRoleEnum	0..1	attr	Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.

Table A.923: StaticSocketConnection

Class	**Std**			
Note	This represents a reference to external standards.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForParagraph*.std			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
date	DateTime	0..1	attr	This element specifies the release date of the external standard if applicable. **Tags:** xml.sequenceOffset=50
position	String	0..1	attr	This represents the reference to the relevant positions of a standard. Kept as a string. **Tags:** xml.sequenceOffset=70
state	String	0..1	attr	This represents version and state of a standard. Kept as a string. **Tags:** xml.sequenceOffset=40
subtitle	String	0..1	attr	This represents the subtitle of the standard. **Tags:** xml.sequenceOffset=30
url	Url	0..1	aggr	This represents the URL of the standard. **Tags:** atp.Splitkey=url xml.sequenceOffset=60

Table A.924: Std

Class	**SubElementMapping**			
Note	This meta-class allows for the definition of mappings of elements of a composite data type.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.subElementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstElement	SubElementRef	0..1	aggr	This represents the first element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=firstElement, firstElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
secondElement	SubElementRef	0..1	aggr	This represents the second element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secondElement, secondElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
textTable Mapping	TextTableMapping	*	aggr	This allows for the text-table translation of individual elements of a composite data type. **Tags:** atp.Splitkey=textTableMapping

Table A.925: SubElementMapping

Class	**SupervisedEntityNeeds**
Note	Specifies the abstract needs on the configuration of the Watchdog Manager for one specific Supervised Entity.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*

▽

△

Class	**SupervisedEntityNeeds**			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
activateAtStart	Boolean	0..1	attr	true/false: supervision activation status of Supervised Entity shall be enabled/disabled at start.
checkpoints	SupervisedEntity CheckpointNeeds	*	ref	This reference indicates the checkpoints belonging to the Supervised Entity. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=checkpoints, checkpoints.supervisedEntity CheckpointNeeds, checkpoints.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
enable Deactivation	Boolean	0..1	attr	true: software-component shall be allowed to deactivate supervision of this SupervisedEntity false: software-component shall be not allowed to deactivate supervision of this SupervisedEntity
expectedAlive Cycle	TimeValue	0..1	attr	Expected cycle time of alive trigger of this Supervised Entity (in seconds).
maxAliveCycle	TimeValue	0..1	attr	Maximum cycle time of alive trigger of this Supervised Entity (in seconds).
minAliveCycle	TimeValue	0..1	attr	Minimum cycle time of alive trigger of this Supervised Entity (in seconds).
toleratedFailed Cycles	PositiveInteger	0..1	attr	Number of consecutive failed alive cycles for this SupervisedEntity which shall be tolerated until the supervision status of the SupervisedEntity is set to WDGM_ALIVE_EXPIRED (see SWS WdgM for more details). Note that this value has to be recalculated with respect to the WdgM's own cycle time for ECU configuration.

Table A.926: SupervisedEntityNeeds

Enumeration	**SupportBufferLockingEnum**
Note	This enumeration represents the ability to define the buffer locking behavior.
Aggregated by	CommunicationBufferLocking.supportBufferLocking
Literal	***Description***
doesNotSupport BufferLocking	Buffer locking is not supported. **Tags:** atp.EnumerationLiteralIndex=0
supportsBuffer Locking	Buffer locking is supported. **Tags:** atp.EnumerationLiteralIndex=1

Table A.927: SupportBufferLockingEnum

Class	**SwAddrMethod**			
Note	Used to assign a common addressing method, e.g. common memory section, to data or code objects. These objects could actually live in different modules or components. **Tags:** atp.recommendedPackage=SwAddrMethods			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**SwAddrMethod**			
memory Allocation KeywordPolicy	MemoryAllocation KeywordPolicyType	0..1	attr	Enumeration to specify the name pattern of the Memory Allocation Keyword.
option	Identifier	*	attr	This attribute introduces the ability to specify further intended properties of the MemorySection in with the related objects shall be placed. These properties are handled as to be selected. The intended options are mentioned in the list. In the Memory Mapping configuration, this option list is used to determine an appropriate MemMapAddressing ModeSet.
section Initialization Policy	SectionInitialization PolicyType	0..1	attr	Specifies the expected initialization of the variables (inclusive those which are implementing VariableData Prototypes). Therefore this is an implementation constraint for initialization code of BSW modules (especially RTE) as well as the start-up code which initializes the memory segment to which the AutosarData Prototypes referring to the SwAddrMethod's are later on mapped. If the attribute is not defined it has the identical semantic as the attribute value "INIT"
sectionType	MemorySectionType	0..1	attr	Defines the type of memory sections which can be associated with this addressing method.

Table A.928: SwAddrMethod

Class	**SwAxisCont**			
Note	This represents the values for the axis of a compound primitive (curve, map). For standard and fix axes, SwAxisCont contains the values of the axis directly. The axis values of SwAxisCont with the category COM_AXIS, RES_AXIS are for display only. For editing and processing, only the values in the related GroupAxis are binding.			
Base	*ARObject*			
Aggregated by	ApplicationValueSpecification.swAxisCont			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	CalprmAxisCategory Enum	0..1	attr	This category specifies the particular axis types: • STD_AXIS • COM_AXIS • RES_AXIS (swArraysize necessary) **Tags:** xml.sequenceOffset=20
swArraysize	ValueList	0..1	aggr	For multidimensional compound primitivies (curve, map ...) it is necessary to know the dimensions.They are specified using swArraySize. • RES_AXIS **Tags:** xml.sequenceOffset=70
swAxisIndex	AxisIndexType	0..1	attr	This property allows to explicitly assign the axis contents to a particular axis. It is specified by numbers where 1 corresponds to the x-axis. It is also possible to derive the axis association from the sequence of the parent. **Tags:** xml.sequenceOffset=50

▽

△

Class	**SwAxisCont**			
swValuesPhys	SwValues	0..1	aggr	swValuesPhys represents the values in the physical domain. **Tags:** xml.sequenceOffset=80
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:** xml.sequenceOffset=30
unitDisplay Name	SingleLanguageUnit Names	0..1	aggr	This represents the display name which is used for the physical unit of the axis. **Tags:** xml.sequenceOffset=40

Table A.929: SwAxisCont

Class	**SwAxisGeneric**			
Note	This meta-class defines a generic axis. In a generic axis the axispoints points are calculated in the ECU. The ECU is equipped with a fixed calculation algorithm. Parameters for the algorithm can be stored in the data component of the ECU. Therefore these parameters are specified in the data declaration, not in the calibration data.			
Base	*ARObject*			
Aggregated by	SwAxisIndividual.swAxisGeneric			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swAxisType	SwAxisType	0..1	ref	Associated axis calculation strategy. **Tags:** xml.sequenceOffset=20
swGenericAxis Param	SwGenericAxisParam	*	aggr	Specific parameter of a generic axis. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table A.930: SwAxisGeneric

Class	**SwAxisGrouped**			
Note	An SwAxisGrouped is an axis which is shared between multiple calibration parameters.			
Base	*ARObject*, *SwCalprmAxisTypeProps*			
Aggregated by	SwCalprmAxis.swCalprmAxisTypeProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
sharedAxisType	ApplicationPrimitive DataType	0..1	ref	This is the datatype of the calibration parameter providing the shared axis.
swAxisIndex	AxisIndexType	0..1	attr	Describes which axis of the referenced calibration parameter provides the values for the group axis. The index satisfies the following convention: • 0 = value axis. in this case, the interpolation result of the referenced parameter is used as a base point index. • The index should only be specified if the parameter under swCalprm contains more than one axis. It is standard practice for the axis index of parameters with more than one axis, to be set to 1, if data has not been assigned to swAxisIndex. **Tags:** xml.sequenceOffset=20

▽

△

Class	**SwAxisGrouped**			
swCalprmRef	SwCalprmRefProxy	1	aggr	This property specifies the calibration parameter which serves as the input axis. In AUTOSAR, the type of the referenced Calibration parameter shall be compatible to the type specified by sharedAxisType. Please note that the multiplicity of this aggregation cannot be set to 0..1 based on the non-mainstream schema generation instructions defined at the aggregation. However, the multiplicity has to be factually considered 0..1 (i.e. a SwAxisGrouped that does not aggregate the role swCalprmRef is still valid according to the XML schema, depending on the use case documented in [constr_1015]). **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.931: SwAxisGrouped

Class	**SwAxisIndividual**			
Note	This meta-class describes an axis integrated into a parameter (field etc.). The integration makes this individual to each parameter. The so-called grouped axis represents the counterpart to this. It is conceived as an independent parameter (see class SwAxisGrouped).			
Base	*ARObject*, *SwCalprmAxisTypeProps*			
Aggregated by	SwCalprmAxis.swCalprmAxisTypeProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuMethod	CompuMethod	0..1	ref	This is the compuMethod which is expected for the axis. It is used in early stages if the particular input-value is not yet available. **Tags:** xml.sequenceOffset=30
dataConstr	DataConstr	0..1	ref	Refers to constraints, e.g. for plausibility checks. **Tags:** xml.sequenceOffset=80
inputVariable Type	ApplicationPrimitive DataType	0..1	ref	This is the datatype of the input value for the axis. This allows to define e.g. a type of curve, where the input value is finalized at the access point.
swAxisGeneric	SwAxisGeneric	0..1	aggr	this specifies the properties of a generic axis if applicable. **Tags:** xml.sequenceOffset=90
swMaxAxis Points	Integer	0..1	attr	Maximum number of base points contained in the axis of a map or curve. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
swMinAxis Points	Integer	0..1	attr	Minimum number of base points contained in the axis of a map or curve. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=70

▽

△

Class	**SwAxisIndividual**			
swVariableRef (ordered)	SwVariableRefProxy	*	aggr	Refers to input variables of the axis. It is possible to specify more than one variable. Here the following is valid: • The variable with the highest priority shall be given first. It is used in the generation of the code and is also displayed first in the application system. • All variables referenced shall be of the same physical nature. This is usually detected in that the conversion formulae affected refer back to the same SI-units. In AUTOSAR this ensured by the constraint, that the referenced input variables shall use a type compatible to "inputVariableType". • This multiple referencing allows a base point distribution for more than one input variable to be used. One example of this are the temperature curves which can depend both on the induction air temperature and the engine temperature. These variables can be displayed simultaneously by MCD systems (adjustment systems), enabling operating points to be shown in the curves. **Tags:** xml.roleElement=false xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
unit	Unit	0..1	ref	This represents the physical unit of the input value of the axis. It is provided to support the case that the particular input variable is not yet known. **Tags:** xml.sequenceOffset=40

Table A.932: SwAxisIndividual

Class	**SwBaseType**			
Note	This meta-class represents a base type used within ECU software. **Tags:** atp.recommendedPackage=BaseTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *BaseType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.933: SwBaseType

Enumeration	**SwCalibrationAccessEnum**
Note	Determines the access rights to a data object w.r.t. measurement and calibration.
Aggregated by	ModeDeclarationGroupPrototype.swCalibrationAccess, SwCalprmAxis.swCalibrationAccess, SwDataDefProps.swCalibrationAccess
Literal	*Description*
notAccessible	The element will not be accessible via MCD tools, i.e. will not appear in the ASAP file. **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**SwCalibrationAccessEnum**
readOnly	The element will only appear as read-only in an ASAP file. **Tags:** atp.EnumerationLiteralIndex=1
readWrite	The element will appear in the ASAP file with both read and write access. **Tags:** atp.EnumerationLiteralIndex=2

Table A.934: SwCalibrationAccessEnum

Class	**SwCalprmAxis**			
Note	This element specifies an individual input parameter axis (abscissa).			
Base	*ARObject*			
Aggregated by	SwCalprmAxisSet.swCalprmAxis			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	CalprmAxisCategoryEnum	0..1	attr	This property specifies the category of a particular axis. **Tags:** xml.sequenceOffset=30
displayFormat	DisplayFormatString	0..1	attr	This property specifies how the axis values shall be displayed e.g. in documents or in measurement and calibration tools. **Tags:** xml.sequenceOffset=100
swAxisIndex	AxisIndexType	0..1	attr	This attribute specifies which axis is specified by the containing SwCalprmAxis. For example in a curve this is usually "1". In a map this is "1" or "2". **Tags:** xml.sequenceOffset=20
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	Describes the applicability of parameters and variables. **Tags:** xml.sequenceOffset=90
swCalprmAxisTypeProps	SwCalprmAxisTypeProps	0..1	aggr	specific properties depending on the type of the axis. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=40 xml.typeElement=true xml.typeWrapperElement=false

Table A.935: SwCalprmAxis

Class	**SwCalprmAxisSet**			
Note	This element specifies the input parameter axes (abscissas) of parameters (and variables, if these are used adaptively).			
Base	*ARObject*			
Aggregated by	SwDataDefProps.swCalprmAxisSet			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swCalprmAxis	SwCalprmAxis	*	aggr	One axis belonging to this SwCalprmAxisSet **Tags:** atp.Splitkey=swCalprmAxis xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.936: SwCalprmAxisSet

Class	SwCalprmRefProxy			
Note	Wrapper class for different kinds of references to a calibration parameter.			
Base	*ARObject*			
Aggregated by	SwAxisGrouped.swCalprmRef, SwDataDependencyArgs.swCalprmRef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arParameter	AutosarParameterRef	0..1	aggr	This represents a Parameter within AUTOSAR. Note that the Datatype of the referenced ParameterDataPrototype shall be an ApplicationDataType of category VALUE.
mcDataInstance	McDataInstance	0..1	ref	This reference is used in the McSupport file to express the final instance of group axis etc. It is not allowed to use this outside of an McDataInstance. The referenced mcDataInstance shall be originated from a ParameterDataPrototype.

Table A.937: SwCalprmRefProxy

Class	SwComponentPrototype			
Note	Role of a software component within a composition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.component			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	SwComponentType	0..1	tref	Type of the instance. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table A.938: SwComponentPrototype

Class	SwComponentPrototypeAssignment			
Note	This meta-class is only required to allow for the variant modeling of an instanceRef. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	CpSoftwareCluster.swComponentAssignment			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swComponent	SwComponent Prototype	0..1	iref	hierarchical tree(s) of Software Components belonging to this CP Software Cluster. This reference is used to describe the belonging SWCs if the CP Software Cluster is described in the context of a System, **Tags:** atp.Splitkey=swComponent, swComponent.context Composition, swComponent.contextComponent, sw Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table A.939: SwComponentPrototypeAssignment

Class	***SwComponentType*** (abstract)
Note	Base class for AUTOSAR software components.
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Subclasses	*AtomicSwComponentType*, CompositionSwComponentType, ParameterSwComponentType
Aggregated by	ARPackage.element

▽

△

Class	***SwComponentType*** (abstract)			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of `ConsistencyNeeds` owned by the enclosing `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consistencyNeeds, consistencyNeeds.shortName, consistencyNeeds.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
port	PortPrototype	*	aggr	The `PortPrototype`s through which this `SwComponentType` can communicate. The aggregation of `PortPrototype` is subject to variability with the purpose to support the conditional existence of `PortPrototype`s. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=port, port.shortName, port.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portGroup, portGroup.shortName, portGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swcMapping Constraint	SwComponentMapping Constraints	*	ref	Reference to constraints that are valid for this SwComponentType. **Tags:** atp.Splitkey=swcMappingConstraint This Attribute is only used by the AUTOSAR Classic Platform.
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentDocumentation, swComponentDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which `UnitGroup`s are relevant in the context of referencing `SwComponentType`. **Tags:** atp.Splitkey=unitGroup This Attribute is only used by the AUTOSAR Classic Platform.

Table A.940: SwComponentType

Class	***SwConnector*** (abstract)
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*
Subclasses	AssemblySwConnector, DelegationSwConnector, PassThroughSwConnector
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector

▽

△

Class	***SwConnector*** (abstract)			
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 `PortInterface`s typing the two `PortPrototype`s which are referenced by the `SwConnector`. **Tags:** atp.Splitkey=mapping

Table A.941: SwConnector

Class	«atpVariation» **SwDataDefProps**			
Note	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. 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. SwDataDefProps covers various aspects: • 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, swAddrMethod, 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, invalid Value • Code generation policy provided by swRecordLayout **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	*ARObject*			
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, Cpp ImplementationDataTypeElement.swDataDefProps, *DataPrototype*.swDataDefProps, DataPrototype TransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, Diagnostic EnvDataElementCondition.swDataDefProps, DiagnosticExtendedDataRecordElement.swDataDefProps, DiagnosticSovdPrimitiveContentElement.swDataDefProps, DltArgumentProps.networkRep, FlatInstance Descriptor.swDataDefProps, ImplementationDataTypeElement.swDataDefProps, InstantiationDataDef Props.swDataDefProps, ISignal.networkRepresentationProps, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.swDataDefProps, *ReceiverComSpec*.network Representation, SecurityEventContextDataElement.networkRepresentation, *SenderComSpec*.network Representation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTarget Props.swDataDefProps, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, System Signal.physicalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
additionalNative TypeQualifier	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

▽

△

Class	«atpVariation» **SwDataDefProps**			
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** atp.Splitkey=annotation 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:** atp.Splitkey=baseType xml.sequenceOffset=50
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:** atp.Splitkey=compuMethod xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:** atp.Splitkey=dataConstr 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
display Presentation	DisplayPresentation Enum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.
implementation DataType	AbstractImplementation DataType	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 • 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:** atp.Splitkey=implementationDataType xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** atp.Splitkey=invalidValue 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.

▽

△

Class	«atpVariation» **SwDataDefProps**			
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:** atp.Splitkey=swAddrMethod xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended typical alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced Sw AddrMethod. **Tags:** xml.sequenceOffset=33
swBit Representation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. **Tags:** atp.Splitkey=swBitRepresentation xml.sequenceOffset=60
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:** xml.sequenceOffset=70
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:** atp.Splitkey=swCalprmAxisSet xml.sequenceOffset=90
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** atp.Splitkey=swComparisonVariable 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:** atp.Splitkey=swDataDependency 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:** atp.Splitkey=swHostVariable xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. **Tags:** xml.sequenceOffset=230

▽

△

Class	«atpVariation» **SwDataDefProps**			
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:** atp.Splitkey=swPointerTargetProps xml.sequenceOffset=280
swRecord Layout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:** atp.Splitkey=swRecordLayout xml.sequenceOffset=290
swRefresh Timing	MultidimensionalTime	0..1	aggr	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. 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. **Tags:** atp.Splitkey=swRefreshTiming xml.sequenceOffset=300
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** atp.Splitkey=swTextProps xml.sequenceOffset=120
swValueBlock Size	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80

▽

△

Class	«atpVariation» **SwDataDefProps**			
swValueBlock SizeMult (ordered)	Numerical	*	attr	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. 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. For one-dimensional value blocks the attribute swValue BlockSize shall be used and this attribute shall not exist. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
unit	Unit	0..1	ref	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. **Tags:** atp.Splitkey=unit xml.sequenceOffset=350
valueAxisData Type	ApplicationPrimitive DataType	0..1	ref	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. **Tags:** atp.Splitkey=valueAxisDataType xml.sequenceOffset=355

Table A.942: SwDataDefProps

Class	**SwGenericAxisParam**			
Note	This meta-class describes a specific parameter of a generic axis. The name of the parameter is defined through a reference to a parameter type defined on a corresponding axis type. The value of the parameter is given here in case that it is not changeable during calibration. Example is shift / offset in a fixed axis.			
Base	*ARObject*			
Aggregated by	SwAxisGeneric.swGenericAxisParam			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swGenericAxis ParamType	SwGenericAxisParam Type	0..1	ref	Parameter type defined on a corresponding axis type. References can only be made to axis parameters types which are defined within the referenced axis type. **Tags:** xml.sequenceOffset=20
vf (ordered)	Numerical	*	attr	This attribute represents the value of the generic axis parameter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false

Table A.943: SwGenericAxisParam

Class	**SwGenericAxisParamType**			
Note	This meta-class describes a generic axis parameter type, namely: • Plausibility checks can be specified via dataConstr. • Textual description (desc), as a formal description is not of any use, due to the large variety of possibilities. • If this parameter contains structures, these can be simulated through the recursive use of SwGeneric AxisParamTypes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SwAxisType.swGenericAxisParamType			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataConstr	DataConstr	0..1	ref	This reference denoted data constraints applicable to the generic axis parameter. **Tags:** atp.Splitkey=dataConstr xml.sequenceOffset=20

Table A.944: SwGenericAxisParamType

Enumeration	**SwImplPolicyEnum**
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Aggregated by	BswInternalTriggeringPoint.swImplPolicy, InternalTriggeringPoint.swImplPolicy, SwDataDefProps.swImplPolicy, Trigger.swImplPolicy
Literal	***Description***
const	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. **Tags:** atp.EnumerationLiteralIndex=0
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 A.945: SwImplPolicyEnum

Class	**SwPointerTargetProps**
Note	This element defines, that the data object (which is specified by the aggregating element) contains a reference to another data object or to a function in the CPU code. This corresponds to a pointer in the C-language. The attributes of this element describe the category and the detailed properties of the target which is either a data description or a function signature.
Base	*ARObject*

▽

△

Class	SwPointerTargetProps			
Aggregated by	SwDataDefProps.swPointerTargetProps			
Attribute	Type	Mult.	Kind	Note
functionPointer Signature	BswModuleEntry	0..1	ref	The referenced BswModuleEntry serves as the signature of a function pointer definition. Primary use case: function pointer passed as argument to other function. **Tags:** atp.Splitkey=functionPointerSignature xml.sequenceOffset=40 This Attribute is only used by the AUTOSAR Classic Platform.
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of the target data type. **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=30
targetCategory	Identifier	0..1	attr	This specifies the category of the target: • In case of a data pointer, it shall specify the category of the referenced data. • In case of a function pointer, it could be used to denote the category of the referenced BswModuleEntry. **Tags:** xml.sequenceOffset=5

Table A.946: SwPointerTargetProps

Class	SwRecordLayout			
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*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
swRecord LayoutGroup	SwRecordLayoutGroup	0..1	aggr	This is the top level record layout group. **Tags:** atp.Splitkey=swRecordLayoutGroup xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.947: SwRecordLayout

Class	SwServiceArg
Note	Specifies the properties of a data object exchanged during the call of an SwService, e.g. an argument or a return value. The SwServiceArg can also be used in the argument list of a C-macro. For this purpose the category shall be set to "MACRO". A reference to implementationDataType can optional be added if the actual argument has an implementationDataType.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	BswModuleEntry.argument, BswModuleEntry.returnType

▽

△

Class	SwServiceArg			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
direction	ArgumentDirection Enum	0..1	attr	Specifies the direction of the data transfer. The direction shall indicate the direction of the actual information that is being consumed by the caller and/or the callee, not the direction of formal arguments in C. The attribute is optional for backwards compatibility reasons. For example, if a pointer is used to pass a memory address for the expected result, the direction shall be "out". If a pointer is used to pass a memory address with content to be read by the callee, its direction shall be "in". **Tags:** xml.sequenceOffset=10
swArraysize	ValueList	0..1	aggr	This turns the argument of the service to an array. **Tags:** atp.Splitkey=swArraysize xml.sequenceOffset=20
swDataDef Props	SwDataDefProps	0..1	aggr	Data properties of this SwServiceArg. **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=30

Table A.948: SwServiceArg

Class	SwSystemconst			
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=SwSystemconsts			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDef Props	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:** atp.Splitkey=swDataDefProps xml.sequenceOffset=40

Table A.949: SwSystemconst

Class	SwSystemconstValue			
Note	This meta-class assigns a particular value to a system constant.			
Base	*ARObject*			
Aggregated by	SwSystemconstantValueSet.swSystemconstantValue			
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:** atp.Splitkey=annotation xml.sequenceOffset=30
swSystemconst	SwSystemconst	0..1	ref	This is the system constant to which the value applies. **Tags:** atp.Splitkey=swSystemconst xml.sequenceOffset=10
value	Numerical	0..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 A.950: SwSystemconstValue

Class	**SwSystemconstantValueSet**			
Note	This meta-class represents the ability to specify a set of system constant values. **Tags:** atp.recommendedPackage=SwSystemconstantValueSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
sw Systemconstant Value	SwSystemconstValue	*	aggr	This is one particular value of a system constant. **Tags:** atp.Splitkey=swSystemconstantValue

Table A.951: SwSystemconstantValueSet

Class	**SwTextProps**			
Note	This meta-class expresses particular properties applicable to strings in variables or calibration parameters.			
Base	*ARObject*			
Aggregated by	SwDataDefProps.swTextProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the semantics of the arraysize for the array representing the string in an `ImplementationDataType`. It is there to support a safe conversion between `ApplicationDataType` and `ImplementationDataType`, even for variable length strings as required e.g. for Support of SAE J1939.

▽

△

Class	**SwTextProps**			
baseType	SwBaseType	0..1	ref	This is the base type of one character in the string. In particular this baseType denotes the intended encoding of the characters in the string on level of ApplicationDataType. **Tags:** atp.Splitkey=baseType xml.sequenceOffset=30
swFillCharacter	Integer	0..1	attr	Filler character for text parameter to pad up to the maximum length swMaxTextSize. The value will be interpreted according to the encoding specified in the associated base type of the data object, e.g. 0x30 (hex) represents the ASCII character zero as filler character and 0 (dec) represents an end of string as filler character. The usage of the fill character depends on the arraySizeSemantics. **Tags:** xml.sequenceOffset=40
swMaxTextSize	Integer	0..1	attr	Specifies the maximum text size in characters. Note the size in bytes depends on the encoding in the corresponding baseType. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table A.952: SwTextProps

Class	**SwValueCont**			
Note	This metaclass represents the content of one particular SwInstance.			
Base	*ARObject*			
Aggregated by	ApplicationValueSpecification.swValueCont			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
swArraysize	ValueList	0..1	aggr	This attribute defines the size of each dimension for compound primitives CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK. For each dimension one value has to be defined, e.g. one in case of COM_AXIS and two or more in case of MAP. **Tags:** atp.Splitkey=swArraysize xml.sequenceOffset=40
swValuesPhys	SwValues	0..1	aggr	swValuesPhys represents the values in the physical domain. **Tags:** atp.Splitkey=swValuesPhys xml.sequenceOffset=50
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:** atp.Splitkey=unit xml.sequenceOffset=20

▽

△

Class	**SwValueCont**			
unitDisplay Name	SingleLanguageUnit Names	0..1	aggr	This specifies how the physical units of the current value set shall be displayed in documents or in user interfaces of tools. **Tags:** atp.Splitkey=unitDisplayName xml.sequenceOffset=30

Table A.953: SwValueCont

Class	«atpMixed» **SwValues**			
Note	This meta-class represents a list of values. These values can either be the input values of a curve (abscissa values) or the associated values (ordinate values). For multidimensional structures, the values are ordered such that they follow the memory layout, see [TPS_SWCT_01882] In particular for maps and cuboids etc. the resulting long value list can be subsectioned using Value Group. But the processing needs to be done as if vg is not there. Note that numerical values and textual values should not be mixed.			
Base	*ARObject*			
Aggregated by	SwAxisCont.swValuesPhys, SwValueCont.swValuesPhys, ValueGroup.vgContents			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
v	Numerical	0..1	attr	This is a non variant Value. It is provided for sake of Compatibility to ASAM CDF. **Tags:** xml.sequenceOffset=40
vf	Numerical	0..1	attr	This allows to specify the value as VariationPoint. It is distinguished to non variant for sake of compatibility to ASAM CDF 2.0. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
vg	ValueGroup	0..1	aggr	This allows to have intersections in the values in order to support specific rendering (eg. using stylesheets). For tools it is important that the v values are always processed in the same (flattened) order and the tool is able to interpret it without respecting vg. **Tags:** xml.sequenceOffset=50
vt	VerbatimString	0..1	attr	This represents the values of textual data elements (Strings). Note that vt uses the | 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. **Tags:** xml.sequenceOffset=30
vtf	NumericalOrText	0..1	aggr	This aggregation represents the ability to provide a value that is either numerical or text which existence is subject to variability. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.954: SwValues

Class	SwVariableRefProxy			
Note	Proxy class for several kinds of references to a variable.			
Base	*ARObject*			
Aggregated by	SwAxisIndividual.swVariableRef, SwDataDefProps.swComparisonVariable, SwDataDefProps.swHostVariable, SwDataDependencyArgs.swVariable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autosarVariable	AutosarVariableRef	0..1	aggr	This represents the reference to a Variable in an Autosar system. Note that the target of the reference within AutosarVariableRef shall be typed by a primitive data type **Tags:** atp.Splitkey=autosarVariable
mcDataInstanceVar	McDataInstance	0..1	ref	This reference is used in the McSupport file to express the final instance of input values etc. It is not allowed to use this outside of an McDataInstance. The referenced mcDataInstance shall be originated from a VariableDataPrototype. **Tags:** atp.Splitkey=mcDataInstanceVar

Table A.955: SwVariableRefProxy

Class	SwcBswMapping			
Note	Maps an SwcInternalBehavior to an BswInternalBehavior. This is required to coordinate the API generation and the scheduling for AUTOSAR Service Components, ECU Abstraction Components and Complex Driver Components by the RTE and the BSW scheduling mechanisms. **Tags:** atp.recommendedPackage=SwcBswMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswBehavior	BswInternalBehavior	0..1	ref	The mapped BswInternalBehavior **Tags:** atp.Splitkey=bswBehavior
runnableMapping	SwcBswRunnableMapping	*	aggr	A mapping between a pair of SWC and BSW runnables. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=runnableMapping, runnableMapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swcBehavior	SwcInternalBehavior	0..1	ref	The mapped SwcInternalBehavior. **Tags:** atp.Splitkey=swcBehavior
synchronizedModeGroup	SwcBswSynchronizedModeGroupPrototype	*	aggr	A pair of SWC and BSW mode group prototypes to be synchronized by the scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=synchronizedModeGroup, synchronizedModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
synchronizedTrigger	SwcBswSynchronizedTrigger	*	aggr	A pair of SWC and BSW Triggers to be synchronized by the scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=synchronizedTrigger, synchronizedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.956: SwcBswMapping

Class	**SwcBswRunnableMapping**			
Note	Maps a BswModuleEntity to a RunnableEntity if it is implemented as part of a BSW module (in the case of an AUTOSAR Service, a Complex Driver or an ECU Abstraction). The mapping can be used by a tool to find relevant information on the behavior, e.g. whether the bswEntity shall be running in interrupt context.			
Base	*ARObject*			
Aggregated by	SwcBswMapping.runnableMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswEntity	BswModuleEntity	0..1	ref	The mapped BswModuleEntity **Tags:** atp.Splitkey=bswEntity
swcRunnable	RunnableEntity	0..1	ref	The mapped SWC runnable. **Tags:** atp.Splitkey=swcRunnable

Table A.957: SwcBswRunnableMapping

Class	**SwcBswSynchronizedModeGroupPrototype**			
Note	Synchronizes a mode group provided by a component via a port with a mode group provided by a BSW module or cluster.			
Base	*ARObject*			
Aggregated by	SwcBswMapping.synchronizedModeGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModeGroup	ModeDeclarationGroup Prototype	0..1	ref	The BSW mode group prototype. **Tags:** atp.Splitkey=bswModeGroup
swcModeGroup	ModeDeclarationGroup Prototype	0..1	iref	The SWC mode group prototype provided by a particular port. **Tags:** atp.Splitkey=swcModeGroup, swcMode Group.contextPPort, swcModeGroup.targetModeGroup **InstanceRef implemented by:** PModeGroupInAtomic SwcInstanceRef

Table A.958: SwcBswSynchronizedModeGroupPrototype

Class	**SwcBswSynchronizedTrigger**			
Note	Synchronizes a Trigger provided by a component via a port with a Trigger provided by a BSW module or cluster.			
Base	*ARObject*			
Aggregated by	SwcBswMapping.synchronizedTrigger			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswTrigger	Trigger	0..1	ref	The BSW Trigger. **Tags:** atp.Splitkey=bswTrigger
swcTrigger	Trigger	0..1	iref	The SWC Trigger provided by a particular port. **Tags:** atp.Splitkey=swcTrigger, swcTrigger.contextPPort, swcTrigger.targetTrigger **InstanceRef implemented by:** PTriggerInAtomicSwc TypeInstanceRef

Table A.959: SwcBswSynchronizedTrigger

Class	**SwcExclusiveAreaPolicy**			
Note	Options how to generate the ExclusiveArea related APIs. If no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply.			
Base	*ARObject*			
Aggregated by	SwcInternalBehavior.exclusiveAreaPolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
apiPrinciple	ApiPrincipleEnum	0..1	attr	Specifies for this ExclusiveArea if either one common set of Enter and Exit APIs for the whole software component is requested from the Rte or if the set of Enter and Exit APIs is expected per RunnableEntity. The default value is "common".
exclusiveArea	ExclusiveArea	0..1	ref	This reference represents the ExclusiveArea for which the policy applies. **Tags:** atp.Splitkey=exclusiveArea

Table A.960: SwcExclusiveAreaPolicy

Class	**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 This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *Implementation*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
behavior	SwcInternalBehavior	0..1	ref	The internal behavior implemented by this Implementation. **Tags:** atp.Splitkey=behavior
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:** atp.Splitkey=perInstanceMemorySize, perInstance MemorySize.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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 A.961: SwcImplementation

Class	**SwcInternalBehavior**
Note	The `SwcInternalBehavior` of an `AtomicSwComponentType` describes the relevant aspects of the software-component with respect to the RTE, i.e. the `RunnableEntity`s and the `RTEEvent`s they respond to.

▽

△

Class	**SwcInternalBehavior**			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *InternalBehavior*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtomicSwComponentType*.internalBehavior, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arTypedPerInstanceMemory	VariableDataPrototype	*	aggr	Defines an AUTOSAR typed memory-block that needs to be available for each instance of the SW-component. This is typically only useful if `supportsMultipleInstantiation` is set to "true" or if the component defines NVRAM access via permanent blocks. 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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory, arTypedPerInstanceMemory.shortName, arTypedPerInstanceMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
event	RTEEvent	*	aggr	This is a `RTEEvent` specified for the particular `SwcInternalBehavior`. The aggregation of `RTEEvent` is subject to variability with the purpose to support the conditional existence of `RTEEvent`s. Note: the number of `RTEEvent`s might vary due to the conditional existence of `PortPrototype`s using `DataReceivedEvent`s or due to different scheduling needs of algorithms. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=event, event.shortName, event.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveAreaPolicy	SwcExclusiveAreaPolicy	*	aggr	Options how to generate the ExclusiveArea related APIs. When no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveAreaPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
explicitInterRunnableVariable	VariableDataPrototype	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitInterRunnableVariable, explicitInterRunnableVariable.shortName, explicitInterRunnableVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
implicitInter Runnable Variable	VariableDataPrototype	*	aggr	Implement state message semantics for establishing communication among runnables of the same component. The aggregation of implicitInterRunnable Variable 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:** atpVariation **Tags:** atp.Splitkey=implicitInterRunnableVariable, implicitInter RunnableVariable.shortName, implicitInterRunnable Variable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The `includedDataTypeSet` is used by a software component for its implementation. **Tags:** atp.Splitkey=includedDataTypeSet
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Tags:** atp.Splitkey=includedModeDeclarationGroupSet
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 Port Prototypes and component local memories like "per InstanceParameter" or "arTypedPerInstanceMemory". **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
perInstance Memory	PerInstanceMemory	*	aggr	Defines a per-instance memory object needed by this software component. The aggregation of PerInstance Memory 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:** atpVariation **Tags:** atp.Splitkey=perInstanceMemory, perInstance Memory.shortName, perInstanceMemory.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
perInstance Parameter	ParameterData Prototype	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=perInstanceParameter, perInstance Parameter.shortName, perInstanceParameter.variation Point.shortLabel vh.latestBindingTime=preCompileTime
portAPIOption	PortAPIOption	*	aggr	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. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portAPIOption, portAPIOption.port, port APIOption.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
runnable	RunnableEntity	*	aggr	This is a `RunnableEntity` specified for the particular `SwcInternalBehavior`. The aggregation of `RunnableEntity` is subject to variability with the purpose to support the conditional existence of `RunnableEntity`s. Note: the number of `RunnableEntity`s might vary due to the conditional existence of `PortPrototype`s using `DataReceivedEvent`s or due to different scheduling needs of algorithms. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=runnable, runnable.shortName, runnable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
service Dependency	SwcService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item. 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. The SwcServiceDependency owned by an SwcInternal Behavior 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>>. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceDependency, service Dependency.shortName, serviceDependency.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

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:** atpVariation **Tags:** atp.Splitkey=sharedParameter, sharedParameter.short Name, sharedParameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
supports Multiple Instantiation	Boolean	0..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. **Tags:** atp.Splitkey=variationPointProxy, variationPoint Proxy.shortName

Table A.962: SwcInternalBehavior

Class	**SwcModeManagerErrorEvent**			
Note	This event is raised when an error occurred during the handling of the referenced ModeDeclarationGroup Prototype.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	This represents the ModeDeclarationGroupPrototype for which this SwcModeManagerErrorEvent is raised in case of an error. **Tags:** atp.Splitkey=modeGroup, modeGroup.context PPort, modeGroup.targetModeGroup **InstanceRef implemented by:** PModeGroupInAtomic SwcInstanceRef

Table A.963: SwcModeManagerErrorEvent

Class	**SwcModeSwitchEvent**			
Note	This event is raised when the specified mode change occurs.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
activation	ModeActivationKind	0..1	attr	Specifies if the event is raised on entering or exiting a specific mode or is raised on the transition between two modes.
mode (ordered)	ModeDeclaration	0..2	iref	The referenced mode or the transition between two modes raises this SwcModeSwitchEvent. **InstanceRef implemented by:** RModeInAtomicSwc InstanceRef

Table A.964: SwcModeSwitchEvent

Class	**SwcServiceDependency**			
Note	Specialization of ServiceDependency in the context of an SwcInternalBehavior. It allows to associate ports, port groups and (in special cases) data defined for an atomic software component to a given ServiceNeeds element.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceDependency*			
Aggregated by	AdaptiveSwcInternalBehavior.serviceDependency, *AtpClassifier*.atpFeature, SwcInternal Behavior.serviceDependency			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime
assignedPort	RoleBasedPort Assignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedPort, assignedPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime
representedPort Group	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeeds and a PortGroup, for example to request a communication mode which applies for communication via these ports. The referred PortGroup shall be local to this atomic SWC, but via the links between the Port Groups, a tool can evaluate this information such that all the ports linked via this port group on the same ECU can be found. **Tags:** atp.Splitkey=representedPortGroup
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds. **Tags:** atp.Splitkey=serviceNeeds, serviceNeeds.short Name

Table A.965: SwcServiceDependency

Class	**SwcToApplicationPartitionMapping**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.swcToApplicationPartitionMapping, SwComponentMapping Constraints.swcToApplicationPartitionMapping, SystemMapping.swcToApplicationPartitionMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Partition	ApplicationPartition	0..1	ref	Reference to an ApplicationPartition to which a Sw ComponentPrototype is mapped. **Tags:** atp.Splitkey=applicationPartition

▽

△

Class	**SwcToApplicationPartitionMapping**			
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. **Tags:** atp.Splitkey=swComponentPrototype, sw ComponentPrototype.contextComposition, swComponent Prototype.contextComponent, swComponent Prototype.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table A.966: SwcToApplicationPartitionMapping

Class	**SwcToEcuMapping**			
Note	This meta-class is used: • to map SwComponentPrototypes to a specific ECU Instance unit, • optionally to map SwComponentPrototypes to a HwElement with category ProcessingUnit, • optionally to map SwComponentPrototypes typed by SensorActuatorSwComponentType to a Hw Element with category SensorActuator. For each combination of ECUInstance and the optional ProcessingUnit and the optional SensorActuator only one SwcToEcuMapping shall be used. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
component	SwComponent Prototype	*	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 aditionally a mapping of some SwComponent Prototype INSIDE the Composition to another ECU Instance the inner mapping overrides the outer mapping. **Tags:** atp.Splitkey=component, component.context Composition, component.contextComponent, component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
ecuInstance	EcuInstance	0..1	ref	Reference to a specific ECU Instance description. **Tags:** atp.Splitkey=ecuInstance

Table A.967: SwcToEcuMapping

Class	**SwcToImplMapping**
Note	Map instances of an AtomicSwComponentType to a specific Implementation. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	SystemMapping.swImplMapping

▽

△

Class	**SwcToImplMapping**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
component	SwComponent Prototype	*	iref	Reference to the software component instances that are being mapped to the specified Implementation. The targeted SwComponentPrototype needs be of the Atomic SwComponentType being implemented by the referenced Implementation. **Tags:** atp.Splitkey=component, component.context Composition, component.contextComponent, component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
component Implementation	SwcImplementation	0..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. **Tags:** atp.Splitkey=componentImplementation

Table A.968: SwcToImplMapping

Class	**SwcToSwcOperationArguments**			
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*			
Aggregated by	CommonSignalPath.operation, ForbiddenSignalPath.operation, PermissibleSignalPath.operation, SeparateSignalPath.operation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
direction	SwcToSwcOperation ArgumentsDirection Enum	0..1	attr	Direction addressed by this SwcToSwcClientServer Operation element.
operation	ClientServerOperation	*	iref	Reference to the operation at the client and at the server side whose arguments are described by SwcToSwc OperationArguments. The two ports referenced shall be connected by a connector in the software component description. **Tags:** atp.Splitkey=operation, operation.context Composition, operation.contextComponent, operation.contextPort, operation.targetOperation **InstanceRef implemented by:** OperationInSystem InstanceRef

Table A.969: SwcToSwcOperationArguments

Class	**SwcToSwcSignal**
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	*ARObject*

▽

△

Class	**SwcToSwcSignal**			
Aggregated by	CommonSignalPath.signal, ForbiddenSignalPath.signal, PermissibleSignalPath.signal, SeparateSignalPath.signal			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	iref	Reference to a data element on the PPortPrototype and to the same data element on the RPortPrototype. **Tags:** atp.Splitkey=dataElement, dataElement.contextComponent, dataElement.contextComposition, dataElement.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeInSystemInstanceRef

Table A.970: SwcToSwcSignal

Enumeration	**SwitchMacAddressLearningEnum**
Note	Defines the MAC address learning mode.
Aggregated by	CouplingElement.switchMacAddressLearningMode
Literal	***Description***
independentVlanLearning	Defines the Independent Vlan Learning (IVL) mode. **Tags:** atp.EnumerationLiteralIndex=1
sharedVlanLearning	Defines the Shared Vlan Learning (SVL) mode. **Tags:** atp.EnumerationLiteralIndex=0

Table A.971: SwitchMacAddressLearningEnum

Class	**SymbolProps**			
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*			
Aggregated by	Allocator.namespace, ApApplicationErrorDomain.namespace, *AtomicSwComponentType*.symbolProps, *CppImplementationDataType*.namespace, ImplementationDataType.symbolProps, *Port Interface*.namespace, SecurityEventDefinition.eventSymbolName, TraceSwitchConfig.namespace			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.972: SymbolProps

Class	SynchronizationTimingConstraint			
Note	This constraint is used to restrict the timing behavior of different, but correlated events or event chains, with regard to synchronization. Two scenarios are supported: • If (synchronizationConstraintType==responseSynchronization) – TimingDescriptionEvents: An arbitrary number of correlated events which play the role of responses shall occur synchronously with respect to a predefined tolerance. – TimingDescriptionEventChains: An arbitrary number of correlated event chains with a common stimulus, but different responses, where the responses shall occur synchronously with respect to a predefined tolerance. • If (synchronizationConstraintType==stimulusSynchronization) – TimingDescriptionEvents:An arbitrary number of correlated events which play the role of stimuli shall occur synchronously with respect to a predefined tolerance. – TimingDescriptionEventChains: An arbitrary number of correlated event chains with a common response, but different stimuli, where the stimuli shall occur synchronously with respect to a predefined tolerance. In case the constraint is imposed on events the following two scenarios are supported: • If (eventOccurrenceKind==singleOccurrence): any of the events shall occur only once in the given time interval. • If (eventOccurrenceKind==multipleOccurrences): any of the events may occur more than once in the given time interval. In other words multiple occurrences of an event within the given time interval are permitted.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingConstraint, Traceable			
Aggregated by	TimingExtension.timingGuarantee, TimingExtension.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
event OccurrenceKind	EventOccurrenceKind Enum	0..1	attr	Indicates whether the referenced events shall occur only once (single occurrence) or multiple times (multiple occurrences) in the given time interval.
scope	TimingDescriptionEvent Chain	*	ref	The event chains that are in the scope of the constraint. Mutually exclusive to scopeEvent, see ([constr_4522]). **Tags:** atp.Splitkey=scope
scopeEvent	TimingDescriptionEvent	*	ref	The events that are in the scope of the constraint. Mutually exclusive to scope, see ([constr_4522]) **Tags:** atp.Splitkey=scopeEvent
synchronization ConstraintType	SynchronizationType Enum	0..1	attr	Indicates whether the associated events of the SynchronizationTimingConstraint have a common stimulus or response.
tolerance	MultidimensionalTime	0..1	aggr	The maximum time interval, within which the synchronized events shall occur. The events may occur in any order within this time interval. The time interval starts at the point-in-time when one of the referenced events occurs. **Tags:** atp.Splitkey=tolerance

Table A.973: SynchronizationTimingConstraint

Class	SynchronousServerCallPoint			
Note	This means that the RunnableEntity is supposed to perform a blocking wait for a response from the server. This Class is only used by the AUTOSAR Classic Platform.			
Base	ARObject, AbstractAccessPoint, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, ServerCallPoint			
Aggregated by	AtpClassifier.atpFeature, RunnableEntity.serverCallPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

△

Class	SynchronousServerCallPoint			
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order. **Tags:** atp.Splitkey=calledFromWithinExclusiveArea

Table A.974: SynchronousServerCallPoint

Class	System			
Note	The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints. 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. **Tags:** atp.recommendedPackage=Systems			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. **Tags:** atp.Splitkey=clientIdDefinitionSet This Attribute is only used by the AUTOSAR Classic Platform.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus. This Attribute is only used by the AUTOSAR Classic Platform.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract. This Attribute is only used by the AUTOSAR Classic Platform.
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:** atp.Splitkey=fibexElement, fibexElement.fibexElement, fibexElement.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolation Routine MappingSet	InterpolationRoutine MappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System. **Tags:** atp.Splitkey=interpolationRoutineMappingSet This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	System			
j1939Shared AddressCluster	J1939SharedAddress Cluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=j1939SharedAddressCluster, j1939Shared AddressCluster.shortName, j1939SharedAddress Cluster.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
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:** atpVariation **Tags:** atp.Splitkey=mapping, mapping.shortName, mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
pncVector Length	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.
rootSoftware Composition	RootSwComposition Prototype	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:** atpVariation **Tags:** atp.Splitkey=rootSoftwareComposition, rootSoftware Composition.shortName, rootSoftware Composition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
swCluster	CpSoftwareCluster	*	ref	CP Software Clusters of this System **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swCluster, swCluster.cpSoftwareCluster, sw Cluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
systemCom SpecDefinition	SystemComSpec DefinitionSet	*	ref	Reference to the set of ComSpec definitions that are used for inter-ECU communication in the System. **Tags:** atp.Splitkey=systemComSpecDefinition

▽

△

Class	**System**			
system Documentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemDocumentation, system Documentation.shortName, system Documentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10 This Attribute is only used by the AUTOSAR Classic Platform.
systemVersion	RevisionLabelString	0..1	attr	Version number of the System Description.

Table A.975: System

Class	**SystemMapping**			
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*			
Aggregated by	System.mapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
application PartitionToEcu Partition Mapping	ApplicationPartitionTo EcuPartitionMapping	*	aggr	Mapping of ApplicationPartitions to EcuPartitions **Stereotypes:** atpVariation **Tags:** atp.Splitkey=applicationPartitionToEcuPartitionMapping, applicationPartitionToEcuPartitionMapping.shortName, applicationPartitionToEcuPartitionMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
appOsTask ProxyToEcu TaskProxy Mapping	AppOsTaskProxyToEcu TaskProxyMapping	*	aggr	Mapping of an OsTaskProxy that was created in the context of a SwComponent to an OsTaskProxy that was created in the context of an Ecu. **Tags:** atp.Splitkey=appOsTaskProxyToEcuTaskProxy Mapping, appOsTaskProxyToEcuTaskProxy Mapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
com Management Mapping	ComManagement Mapping	*	aggr	Mappings between Mode Management PortGroups and communication channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=comManagementMapping, comManagement Mapping.shortName, comManagementMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**SystemMapping**			
cryptoService Mapping	CryptoServiceMapping	*	aggr	This aggregation represents the collection of crypto service mappings in the context of the enclosing System Mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=cryptoServiceMapping, cryptoService Mapping.shortName, cryptoServiceMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
cyclicHandling ComDataToOs TaskProxy Mapping	CyclicHandlingCom DataToOsTaskProxy Mapping	*	aggr	Mapping of VariableDataPrototypes to an OsTaskProxy for the Cyclic Handling of Communication Data in the RTE. **Tags:** atp.Splitkey=cyclicHandlingComDataToOsTask ProxyMapping, cyclicHandlingComDataToOsTaskProxy Mapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
dataMapping	DataMapping	*	aggr	The data mappings defined. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataMapping, dataMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
ddsISignalTo TopicMapping	DdsCpISignalToDds TopicMapping	*	aggr	Collection of DdsISignalToDdsTopicMappings. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ddsISignalToTopicMapping, ddsISignalTo TopicMapping.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
ecuPartitionTo CoreMapping	EcuPartitionToCore Mapping	*	aggr	Mapping of EcuPartions to a Cores. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuPartitionToCoreMapping, ecuPartitionTo CoreMapping.shortName, ecuPartitionToCore Mapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
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:** atp.Splitkey=ecuResourceMapping, ecuResource Mapping.shortName, ecuResourceMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**SystemMapping**			
j1939Controller ApplicationTo J1939Node Mapping	J1939Controller ApplicationToJ1939 NodeMapping	*	aggr	Mapping of a J1939ControllerApplication to a J1939Node. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=j1939ControllerApplicationToJ1939Node Mapping, j1939ControllerApplicationToJ1939Node Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
mapping Constraint	MappingConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of SW components to ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mappingConstraint, mapping Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
pncMapping	PncMapping	*	aggr	Mappings between Virtual Function Clusters and Partial Network Clusters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pncMapping, pncMapping.ident.shortName, pncMapping.pncIdentifier, pncMapping.shortLabel, pnc Mapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
portElementTo ComResource Mapping	PortElementTo Communication ResourceMapping	*	aggr	maps a communication resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portElementToComResourceMapping, port ElementToComResourceMapping.shortName, port ElementToComResourceMapping.variationPoint.short Label vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
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:** atp.Splitkey=resourceEstimation, resource Estimation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**SystemMapping**			
resourceTo Application Partition Mapping	CpSoftwareCluster ResourceToApplication PartitionMapping	*	aggr	Maps a Software Cluster resource to an Application Partition to restrict the usage. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=resourceToApplicationPartitionMapping, resourceToApplicationPartitionMapping.shortName, resourceToApplicationPartitionMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
rteEvent Separation	RteEventInSystem Separation	*	aggr	Separation constraint that limits the mapping freedom for the mapping of RteEvents to OsTasks in the System context. **Tags:** atp.Splitkey=rteEventSeparation, rteEvent Separation.shortName This Attribute is only used by the AUTOSAR Classic Platform.
rteEventToOs TaskProxy Mapping	RteEventInSystemToOs TaskProxyMapping	*	aggr	Constraint that enforces a mapping of RteEvent to a particular OsTask in the System context. **Tags:** atp.Splitkey=rteEventToOsTaskProxyMapping, rte EventToOsTaskProxyMapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
signalPath Constraint	SignalPathConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of data elements to signals. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=signalPathConstraint, signalPath Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
softwareCluster ToApplication Partition Mapping	CpSoftwareClusterTo ApplicationPartition Mapping	*	aggr	The mapping of ApplicationPartitions to a CpSoftware Cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=softwareClusterToApplicationPartition Mapping, softwareClusterToApplicationPartition Mapping.shortName, softwareClusterToApplication PartitionMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
softwareCluster ToResource Mapping	CpSoftwareClusterTo ResourceMapping	*	aggr	maps a service resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=softwareClusterToResourceMapping, softwareClusterToResourceMapping.shortName, softwareClusterToResourceMapping.variationPoint.short Label vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**SystemMapping**			
swCluster Mapping	CpSoftwareClusterTo EcuInstanceMapping	*	aggr	The mappings of SW cluster to ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swClusterMapping, swClusterMapping.short Name, swClusterMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
swcTo Application Partition Mapping	SwcToApplication PartitionMapping	*	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:** atpVariation **Tags:** atp.Splitkey=swcToApplicationPartitionMapping, swcTo ApplicationPartitionMapping.shortName, swcTo ApplicationPartitionMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
swImplMapping	SwcToImplMapping	*	aggr	The mappings of AtomicSoftwareComponent Instances to Implementations. atpVariation: Derived, because SwcToEcuMapping is variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swImplMapping, swImplMapping.shortName, swImplMapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
swMapping	SwcToEcuMapping	*	aggr	The mappings of SW components to ECUs. atpVariation: SWC shall be mapped to other ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swMapping, swMapping.shortName, sw Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
systemSignal GroupToCom Resource Mapping	SystemSignalGroupTo Communication ResourceMapping	*	aggr	Mapping of a communication resource to a SystemSignal Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemSignalGroupToComResource Mapping, systemSignalGroupToComResource Mapping.shortName, systemSignalGroupToCom ResourceMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	SystemMapping			
systemSignalTo ComResource Mapping	SystemSignalTo Communication ResourceMapping	*	aggr	Mapping of a communication resource to a SystemSignal. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemSignalToComResourceMapping, systemSignalToComResourceMapping.shortName, systemSignalToComResourceMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

Table A.976: SystemMapping

Class	SystemSignal			
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*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicLength	Boolean	0..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. **Tags:** atp.Splitkey=physicalProps

Table A.977: SystemSignal

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

Table A.978: SystemSignalGroup

Class	SystemSignalGroupToCommunicationResourceMapping			
Note	This meta class maps a communication resource to a SystemSignalGroup. This mapping can be used in an early process stage in which the DataMapping linking the Ports and mapped CpSoftwareCluster CommunicationResource(s) to SystemSignals of a SystemSignalGroup is not yet available. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.systemSignalGroupToComResourceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
softwareCluster ComResource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=softwareClusterComResource
systemSignal Group	SystemSignalGroup	0..1	ref	SystemSignalGroup to which the communication resource is assigned **Tags:** atp.Splitkey=systemSignalGroup

Table A.979: SystemSignalGroupToCommunicationResourceMapping

Class	SystemSignalToCommunicationResourceMapping			
Note	This meta class maps a communication resource to a SystemSignal. This mapping can be used in an early process stage in which the DataMapping linking the Ports and mapped CpSoftwareCluster CommunicationResource(s) to the SystemSignal is not yet available. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.systemSignalToComResourceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
softwareCluster ComResource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=softwareClusterComResource
systemSignal	SystemSignal	0..1	ref	SystemSignal to which the communication resource is assigned **Tags:** atp.Splitkey=systemSignal

Table A.980: SystemSignalToCommunicationResourceMapping

Class	SystemTiming			
Note	A model element used to refine timing descriptions and constraints (from a VfbTiming) at System level, utilizing information about topology, software deployment, and signal mapping described in the System Template. TimingDescriptions aggregated by SystemTiming are restricted to events which are derived from the class TDEventVfb, TDEventSwcInternalBehavior and TDEventCom. **Tags:** atp.recommendedPackage=TimingExtensions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
system	System	0..1	ref	This defines the scope of a SystemTiming. All corresponding timing descriptions and constraints shall be defined within this scope. **Tags:** atp.Splitkey=system

Table A.981: SystemTiming

Class	**TDCpSoftwareClusterMapping**			
Note	This is used to specify a mapping between a software cluster that provides temporal and dynamic resources and the software clusters that need these resources. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterToTdMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provider	CpSoftwareCluster	0..1	ref	This is the software cluster that provides the temporal and dynamic resource. **Tags:** atp.Splitkey=provider
requestor	CpSoftwareCluster	*	ref	This is the software cluster that requests the temporal and dynamic resource. **Tags:** atp.Splitkey=requestor
timing Description	TimingDescription	0..1	ref	The timing description representing the temporal and dynamic resource. **Tags:** atp.Splitkey=timingDescription

Table A.982: TDCpSoftwareClusterMapping

Class	**TDCpSoftwareClusterResourceMapping**			
Note	This is used to assign an unequivocal global resource identification to a temporal and dynamic resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterResourceToTdMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
resource	CpSoftwareCluster Resource	0..1	ref	The specific resource identification assigned to the temporal and dynamic resource. **Tags:** atp.Splitkey=resource
timing Description	TimingDescription	0..1	ref	The timing description representing the temporal and dynamic resource. **Tags:** atp.Splitkey=timingDescription

Table A.983: TDCpSoftwareClusterResourceMapping

Class	***TDEventBsw*** (abstract)			
Note	This is used to describe timing events related to BSW modules. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	TDEventBswModeDeclaration, TDEventBswModule			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModule Description	BswModuleDescription	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=bswModuleDescription

Table A.984: TDEventBsw

Class	**TDEventBswInternalBehavior**			
Note	This is used to describe timing events related to the BswInternalBehavior of a BSW module. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModule Entity	BswModuleEntity	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=bswModuleEntity
tdEventBsw Internal BehaviorType	TDEventBswInternal BehaviorTypeEnum	0..1	attr	The specific type of this timing event.

Table A.985: TDEventBswInternalBehavior

Class	**TDEventBswModeDeclaration**			
Note	This is used to describe timing events related to the mode communication on BSW level. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventBsw*, *TimingDescription*, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
entryMode Declaration	ModeDeclaration	0..1	ref	Optional parameter which refines the scope of the TDEventBswModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall enter into the referenced ModeDeclaration. **Tags:** atp.Splitkey=entryModeDeclaration
exitMode Declaration	ModeDeclaration	0..1	ref	Optional parameter which refines the scope of the TDEventBswModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall exit from the referenced ModeDeclaration. **Tags:** atp.Splitkey=exitModeDeclaration
mode Declaration	ModeDeclarationGroup Prototype	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=modeDeclaration
tdEventBsw Mode DeclarationType	TDEventBswMode DeclarationTypeEnum	0..1	attr	The specific type of this timing event.

Table A.986: TDEventBswModeDeclaration

Class	**TDEventBswModule**			
Note	This is used to describe timing events related to the interaction between BSW modules. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventBsw*, *TimingDescription*, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswModule Entry	BswModuleEntry	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=bswModuleEntry
tdEventBsw ModuleType	TDEventBswModule TypeEnum	0..1	attr	The specific type of this timing event.

Table A.987: TDEventBswModule

Class	TDEventComplex			
Note	This is used to describe complex timing events. The context of a complex timing event either is described informally, e.g. using the documentation block, or is described formally by the associated TDEventOccurrenceExpression.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.988: TDEventComplex

Class	*TDEventCycleStart* (abstract)			
Note	This is the abstract parent class to describe timing events related to a point in time where a communication cycle starts. Via the attribute "cycleRepetition", a filtered view to the cycle start can be defined.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	TDEventFrClusterCycleStart			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cycleRepetition	Integer	0..1	attr	The start of every <cycleRepetition> cycle is targeted by this event.

Table A.989: TDEventCycleStart

Class	TDEventFrClusterCycleStart			
Note	This is used to describe the timing event related to a point in time where a communication cycle starts on a FlexRay cluster.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TDEventCycleStart*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frCluster	FlexrayCluster	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=frCluster

Table A.990: TDEventFrClusterCycleStart

Class	TDEventFrame			
Note	This is used to describe timing events related to the exchange of frames between the communication controller and the bus specific (FlexRay / CAN / LIN) Interface BSW module.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frame	Frame	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=frame
physical Channel	PhysicalChannel	0..1	ref	The PhysicalChannel on which the Frame is transmitted. **Tags:** atp.Splitkey=physicalChannel
tdEventType	TDEventFrameType Enum	0..1	attr	The specific type of this timing event.

Table A.991: TDEventFrame

Class	TDEventFrameEthernet			
Note	This is used to describe timing description events related to the exchange of Ethernet frames between an Ethernet communication controller and the BSW Ethernet interface and driver module.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
staticSocketConnection	StaticSocketConnection	0..1	ref	Specifies the SocketConnection by the means of which Physical Data Units (PDU) are transmitted or received within an Ethernet Frame. **Tags:** atp.Splitkey=staticSocketConnection
tdEventType	TDEventFrameEthernetTypeEnum	0..1	attr	This is used to describe the specific event type of a TDEventFrameEthernet.
tdHeaderIdFilter	TDHeaderIdRange	*	aggr	Specifies the header identifier or a range of header identifiers that if contained in the Ethernet frame let the TDEventFrameEthernet occur. **Tags:** atp.Splitkey=tdHeaderIdFilter
tdPduTriggeringFilter	PduTriggering	*	ref	Specifies the PDU that if contained in the Ethernet frame let the TDEventFrameEthernet occur. **Tags:** atp.Splitkey=tdPduTriggeringFilter

Table A.992: TDEventFrameEthernet

Class	TDEventIPdu			
Note	Describe timing events related to the exchange of `IPdu`s between the bus specific (FlexRay / CAN / LIN) Interface BSW module and COM.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iPdu	IPdu	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=iPdu
physicalChannel	PhysicalChannel	0..1	ref	The PhysicalChannel on which the IPdu is transmitted. **Tags:** atp.Splitkey=physicalChannel
tdEventType	TDEventIPduTypeEnum	0..1	attr	The specific type of this timing event.

Table A.993: TDEventIPdu

Class	TDEventISignal			
Note	Describe timing events related to the exchange of `TDEventISignal`s between `COM` and `RTE`.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iSignal	ISignal	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=iSignal
physicalChannel	PhysicalChannel	0..1	ref	The PhysicalChannel on which the ISignal is transmitted. **Tags:** atp.Splitkey=physicalChannel
tdEventType	TDEventISignalTypeEnum	0..1	attr	The specific type of this timing event.

Table A.994: TDEventISignal

Class	TDEventModeDeclaration			
Note	A TimingDescriptionEvent triggered by the initiation/propagation of a mode switch in a ModeSwitchInterface on VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
entryMode Declaration	ModeDeclaration	0..1	ref	Optional parameter which refines the scope of the TDEventModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall enter into the referenced ModeDeclaration. **Tags:** atp.Splitkey=entryModeDeclaration
exitMode Declaration	ModeDeclaration	0..1	ref	Optional parameter which refines the scope of the TDEventModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall exit from the referenced ModeDeclaration. **Tags:** atp.Splitkey=exitModeDeclaration
mode Declaration	ModeDeclarationGroup Prototype	0..1	ref	The referenced ModeDeclarationGroupPrototype from a{ModeSwitchInterface}]. **Tags:** atp.Splitkey=modeDeclaration
tdEventMode DeclarationType	TDEventMode DeclarationTypeEnum	0..1	attr	The specific type of this timing event.

Table A.995: TDEventModeDeclaration

Class	TDEventOccurrenceExpression			
Note	This is used to specify a filter on the occurrences of TimingDescriptionEvents by means of a TDEventOccurrenceExpressionFormula. Filter criteria can be variable and argument values, i.e. the timing event only occurs for specific values, as well as the temporal characteristics of the occurrences of arbitrary timing events.			
Base	*ARObject*			
Aggregated by	*TimingDescriptionEvent*.occurrenceExpression			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument	AutosarOperation ArgumentInstance	*	aggr	An occurrence expression can reference an arbitrary number of ArgumentDataPrototypes in its expression. This association aggregates instance references to OperationArgumentPrototypes which can be referenced in the expression. **Tags:** atp.Splitkey=argument, argument.shortName
formula	TDEventOccurrence ExpressionFormula	0..1	aggr	This is the expression formula which is used to describe the occurrence expression. **Tags:** atp.Splitkey=formula
function	TDEventOccurrence ExpressionFunction Enum	0..1	attr	As per TDEventOccurrenceExpressionFunctionEnum
mode	TimingModeInstance	*	aggr	An occurrence expression can reference an arbitrary number of TimingModeInstances in its expression. This association aggregates instance references to Mode Declaration which can be referenced in the expression. **Tags:** atp.Splitkey=mode, mode.shortName



△

Class	**TDEventOccurrenceExpression**			
variable	AutosarVariable Instance	*	aggr	An occurrence expression can reference an arbitrary number of `VariableDataPrototype`s in its expression. This association aggregates instance references to `VariableDataPrototype`s which can be referenced in the expression. **Tags:** atp.Splitkey=variable, variable.shortName

Table A.996: TDEventOccurrenceExpression

Class	**TDEventOperation**			
Note	A `TimingDescriptionEvent` triggered by the sending/receiving or invocation/return of a `ClientServerOperation` in a `ClientServerInterface` on VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	ref	The referenced `ClientServerOperation` from a `ClientServerInterface`. **Tags:** atp.Splitkey=operation
tdEvent OperationType	TDEventOperationType Enum	0..1	attr	The specific type of this timing event.

Table A.997: TDEventOperation

Class	**TDEventSwcInternalBehavior**			
Note	This is used to describe timing events related to the SwcInternalBehavior of an AtomicSwComponent Type. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventSwc*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
runnable	RunnableEntity	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=runnable
tdEventSwc Internal BehaviorType	TDEventSwcInternal BehaviorTypeEnum	0..1	attr	The specific type of this timing event.
variableAccess	VariableAccess	0..1	ref	The scope of this timing event. **Tags:** atp.Splitkey=variableAccess

Table A.998: TDEventSwcInternalBehavior

Class	**TDEventSwcInternalBehaviorReference**
Note	This is used to reference timing description events related to the Software Component (SW-C) view which are specified in other timing views. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventSwc*, *TimingDescription*, *TimingDescriptionEvent*
Aggregated by	*TimingExtension*.timingDescription

▽

△

Class	**TDEventSwcInternalBehaviorReference**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
referenced TDEventSwc	TDEventSwc	0..1	ref	The referenced timing description event. **Tags:** atp.Splitkey=referencedTDEventSwc

Table A.999: TDEventSwcInternalBehaviorReference

Enumeration	**TDEventSwcInternalBehaviorTypeEnum**
Note	This is used to describe the specific event type of a TDEventSwcInternalBehavior. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType
Literal	***Description***
runnableEntity Activated	A point in time where the associated RunnableEntity has been activated, which means that it has entered the state "to be started". **Tags:** atp.EnumerationLiteralIndex=0
runnableEntity Started	A point in time where the associated RunnableEntity has entered the state "started" after its activation. **Tags:** atp.EnumerationLiteralIndex=1
runnableEntity Terminated	A point in time where the associated RunnableEntity has terminated and entered the state "suspended". **Tags:** atp.EnumerationLiteralIndex=2
runnableEntity VariableAccess	A point in time where the associated variable is accessed. **Tags:** atp.EnumerationLiteralIndex=3

Table A.1000: TDEventSwcInternalBehaviorTypeEnum

Class	**TDEventTrigger**			
Note	A `TimingDescriptionEvent` triggered by the activation and release of a `Trigger` in a `TriggerInterface` on VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tdEventTriggerType	TDEventTriggerTypeEnum	0..1	attr	The specific type of this timing event.
trigger	Trigger	0..1	ref	The referenced `Trigger` from a `TriggerInterface`. **Tags:** atp.Splitkey=trigger

Table A.1001: TDEventTrigger

Enumeration	**TDEventTriggerTypeEnum**
Note	This is used to describe the specific event type of a TDEventTrigger.
Aggregated by	TDEventTrigger.tdEventTriggerType
Literal	***Description***
triggerActivated	A point in time where the referenced trigger has been successfully released and is activating runnable entities of the receiving SW-C. **Tags:** atp.EnumerationLiteralIndex=0
triggerReleased	A point in time where the referenced trigger has been successfully released by the emitting SW-C. **Tags:** atp.EnumerationLiteralIndex=1

Table A.1002: TDEventTriggerTypeEnum

Class	**TDEventVariableDataPrototype**			
Note	A `TimingDescriptionEvent` triggered by the sending/receiving of a `VariableDataPrototype` in a `SenderReceiverInterface` on VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	0..1	ref	The referenced `VariableDataPrototype` from a `SenderReceiverInterface`. **Tags:** atp.Splitkey=dataElement
tdEventVariableDataPrototypeType	TDEventVariableDataPrototypeTypeEnum	0..1	attr	The specific type of this timing event.

Table A.1003: TDEventVariableDataPrototype

Enumeration	**TDEventVariableDataPrototypeTypeEnum**
Note	This is used to describe the specific event type of a TDEventVariableDataPrototype
Aggregated by	TDEventVariableDataPrototype.tdEventVariableDataPrototypeType
Literal	***Description***
variableDataPrototypeReceived	A point in time where the referenced variable data prototype has been successfully transmitted and is available in the related communication buffer (of the RTE) for the receiving SWC. **Tags:** atp.EnumerationLiteralIndex=0
variableDataPrototypeSent	A point in time where the referenced variable data prototype has been successfully sent out by the sending SWC, so that it is available in the related communication buffer (of the RTE) for transmission. **Tags:** atp.EnumerationLiteralIndex=1

Table A.1004: TDEventVariableDataPrototypeTypeEnum

Class	***TDEventVfb*** (abstract)			
Note	A `TimingDescriptionEvent` occuring on a Virtual Functional Bus (VFB) `PortPrototype`.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	*TDEventVfbPort*, TDEventVfbPortGroup, TDEventVfbReference			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.1005: TDEventVfb

Class	***TDEventVfbPort*** (abstract)			
Note	A `TimingDescriptionEvent` occuring on a `PortPrototype`.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	TDEventModeDeclaration, TDEventOperation, TDEventTrigger, TDEventVariableDataPrototype			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
isExternal	Boolean	0..1	attr	This attribute is used to refer to external events that are related to hardware I/O, like physical sensors and actuators, at Virtual Functional Bus (VFB) level. This Attribute is only used by the AUTOSAR Classic Platform.
portPrototype	PortPrototype	0..1	iref	`PortPrototype` on which the TimingEvent occurs **Tags:** atp.Splitkey=portPrototype, portPrototype.contextComponent, portPrototype.targetPPort, portPrototype.targetRPort **InstanceRef implemented by:** PortInCompositionTypeInstanceRef
portPrototypeBlueprint	PortPrototypeBlueprint	0..1	ref	port on which the TimingEvent shall apply (in the context of an AUTOSAR blueprint) **Tags:** atp.Splitkey=portPrototypeBlueprint

Table A.1006: TDEventVfbPort

Class	**TDEventVfbReference**			
Note	Reference to "other" `TimingDescriptionEvent`s. These other `TimingDescriptionEvent`s may be specified in other `TimingExtension`s (Timing Views) and re-used in this view.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TimingDescription*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
referencedTDEventVfb	TDEventVfb	0..1	ref	The referenced timing description event. **Tags:** atp.Splitkey=referencedTDEventVfb

Table A.1007: TDEventVfbReference

Class	**TDHeaderIdRange**			
Note	Specifies a range of PDU header identifiers. This range is specified by a minimum and maximum header identifier; and the maximum header identifier shall be greater than or equal the minimum header identifier.			
Base	*ARObject*			
Aggregated by	TDEventFrameEthernet.tdHeaderIdFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxHeaderId	Integer	0..1	attr	Specifies the maximum PDU header identifier, in other words the upper bound of a range of PDU header identifiers.
minHeaderId	Integer	0..1	attr	Specifies the minimum PDU header identifier, in other words the lower bound of a range of PDU header identifiers.

Table A.1008: TDHeaderIdRange

Class	TargetIPduRef			
Note	Target destination of the referencing mapping.			
Base	*ARObject*			
Aggregated by	IPduMapping.targetIPdu			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defaultValue	PduMappingDefaultValue	0..1	aggr	If no I-Pdu has been received a default value will be distributed. **Tags:** atp.Splitkey=defaultValue
targetIPdu	PduTriggering	0..1	ref	IPdu Reference **Tags:** atp.Splitkey=targetIPdu

Table A.1009: TargetIPduRef

Class	TcpIpIcmpv4Props			
Note	This meta-class specifies the configuration options for ICMPv4 (Internet Control Message Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpIcmpProps.icmpV4Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpIcmpV4EchoReplyEnabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcpIpIcmpV4Ttl	PositiveInteger	0..1	attr	This attribute is only relevant in case that ICMP (Internet Control Message Protocol) is used. It specifies the default Time-to-live value of outgoing ICMP packets.

Table A.1010: TcpIpIcmpv4Props

Class	TcpIpIcmpv6Props			
Note	This meta-class specifies the configuration options for ICMPv6 (Internet Control Message Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpIcmpProps.icmpV6Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpIcmpV6EchoReplyAvoidFragmentation	Boolean	0..1	attr	This attribute defines whether the echo reply is only transmitted in case that the incoming ICMPv6 Echo Request (Pings) fits the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation.
tcpIpIcmpV6EchoReplyEnabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcpIpIcmpV6HopLimit	PositiveInteger	0..1	attr	Default Hop-Limit value of outgoing ICMPv6 packets.
tcpIpIcmpV6MsgDestinationUnreachableEnabled	Boolean	0..1	attr	This attribute Enables/Disables the transmission of Destination Unreachable Messages.
tcpIpIcmpV6MsgParameterProblemEnabled	Boolean	0..1	attr	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.

Table A.1011: TcpIpIcmpv6Props

Class	**TcpOptionFilterList**			
Note	Permitted list for the filtering of TCP options. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TcpOptionFilterSet.tcpOptionFilterList			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowedTcp Option	PositiveInteger	*	attr	TCP option kind allowed by this filter.

Table A.1012: TcpOptionFilterList

Class	**TcpProps**			
Note	This meta-class specifies the configuration options for TCP (Transmission Control Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpProps.tcpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpCongestion Avoidance Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.
tcpDelayedAck Timeout	TimeValue	0..1	attr	The maximal time an acknowledgement is delayed for transmission in seconds.
tcpFast Recovery Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.
tcpFast Retransmit Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.
tcpFinWait2 Timeout	TimeValue	0..1	attr	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.
tcpKeepAlive Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6.
tcpKeepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
tcpKeepAlive ProbesMax	PositiveInteger	0..1	attr	Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.
tcpKeepAlive Time	TimeValue	0..1	attr	Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe.
tcpMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if tcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.
tcpMsl	TimeValue	0..1	attr	Maximum segment lifetime in [s].
tcpNagle Enabled	Boolean	0..1	attr	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 per Socket (with the attribute TcpTp.nagle Algorithm).
tcpReceive WindowMax	PositiveInteger	0..1	attr	Default value of maximum receive window in bytes.

▽

△

Class	**TcpProps**			
tcp Retransmission Timeout	TimeValue	0..1	attr	Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled, no TCP segments shall be retransmitted.
tcpSlowStart Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681.
tcpSynMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP SYN is retransmitted.
tcpSynReceived Timeout	TimeValue	0..1	attr	Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgement after having both received and sent a connection request.
tcpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing TCP packets.

Table A.1013: TcpProps

Class	**TcpTp**			
Note	Content Model for TCP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
keepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
keepAlive ProbesMax	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 sent.
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.
receiveWindow Min	PositiveInteger	0..1	attr	Minimum size of the TCP receive window in bytes.
tcp Retransmission Timeout	TimeValue	0..1	attr	Defines the timeout in seconds before an unacknowledged TCP segment is sent again. If the tcp RetransmissionTimeout is not defined or set to "INF", no TCP segments shall be re-transmitted.
tcpTpPort	TpPort	0..1	aggr	TCP Port configuration. **Tags:** atp.Splitkey=tcpTpPort

Table A.1014: TcpTp

Class	**TextTableMapping**			
Note	Defines the mapping of two `DataPrototype`s typed by `AutosarDataType`s that refer to `CompuMethod`s of `category` `TEXTTABLE`, `SCALE_LINEAR_AND_TEXTTABLE` or `BITFIELD_TEXTTABLE`.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.textTableMapping, SenderRecArrayTypeMapping.senderToSignalTextTable Mapping, SenderRecArrayTypeMapping.signalToReceiverTextTableMapping, SenderReceiverToSignal Mapping.senderToSignalTextTableMapping, SenderReceiverToSignalMapping.signalToReceiverTextTable Mapping, SenderRecRecordElementMapping.senderToSignalTextTableMapping, SenderRecRecord ElementMapping.signalToReceiverTextTableMapping, SubElementMapping.textTableMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**TextTableMapping**			
bitfieldTextTable MaskFirst	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the first element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
bitfieldTextTable MaskSecond	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the second element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
identical Mapping	Boolean	0..1	attr	If identicalMapping is set == true the values of the two referenced DataPrototypes do not need any conversion of the values.
mapping Direction	MappingDirectionEnum	0..1	attr	Specifies the conversion direction for which the TextTable Mapping is applicable.
valuePair	TextTableValuePair	*	aggr	Defines a pair of values which are translated into each other. **Tags:** atp.Splitkey=valuePair

Table A.1015: TextTableMapping

Class	**TextTableValuePair**			
Note	Defines a pair of text values which are translated into each other.			
Base	*ARObject*			
Aggregated by	TextTableMapping.valuePair			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstValue	Numerical	0..1	attr	Value of first DataPrototype provided similar to 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. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
secondValue	Numerical	0..1	attr	Value of second DataPrototype provided similar to 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. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.1016: TextTableValuePair

Class	**TextValueSpecification**			
Note	The purpose of TextValueSpecification is to define the labels that correspond to enumeration values.			
Base	*ARObject*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, MetaDataItem.metaDataItemType, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	VerbatimString	0..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 A.1017: TextValueSpecification

Class	**TimeRangeType**			
Note	The timeRange can be specified with the value attribute. Optionally a tolerance can be defined.			
Base	*ARObject*			
Aggregated by	CyclicTiming.timeOffset, CyclicTiming.timePeriod, EventControlledTiming.repetitionPeriod			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tolerance	TimeRangeType Tolerance	0..1	aggr	Optional specification of a tolerance. **Tags:** atp.Splitkey=tolerance
value	TimeValue	0..1	attr	Average value of a date (in seconds)

Table A.1018: TimeRangeType

Class	***TimingDescription*** (abstract)			
Note	The abstract parent class of the model elements that are used to define the scope of a timing constraint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*TimingDescriptionEvent*, TimingDescriptionEventChain			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.1019: TimingDescription

Class	***TimingDescriptionEvent*** (abstract)
Note	A timing event is the abstract representation of a specific system behavior -- that can be observed at runtime -- in the AUTOSAR specification. Timing events are used to define the scope for timing constraints. Depending on the specific scope, the view on the system, and the level of abstraction different types of events are defined. In order to avoid confusion with existing event descriptions in the AUTOSAR templates the timing specific event types use the prefix TD.

▽

△

Class	**TimingDescriptionEvent** (abstract)			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*			
Subclasses	*TDEventBsw*, TDEventBswInternalBehavior, *TDEventCom*, TDEventComplex, *TDEventSwc*, *TDEventVfb*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clockReference	TimingClock	0..1	ref	Optional reference to a clock that holds the time base for an TD event. **Tags:** atp.Splitkey=clockReference
occurrence Expression	TDEventOccurrence Expression	0..1	aggr	The occurrence expression for this `TimingDescriptionEvent` **Tags:** atp.Splitkey=occurrenceExpression

Table A.1020: TimingDescriptionEvent

Class	**TimingDescriptionEventChain**			
Note	An event chain describes the causal order for a set of functionally dependent timing events. Each event chain has a well defined stimulus and response, which describe its start and end point. Furthermore, it can be hierarchically decomposed into an arbitrary number of sub-chains, so called *event chain segments*.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
isPipelining Permitted	Boolean	0..1	attr	States whether the scheduled entities in an LET interval shall use pipelined execution or not i.e. "permitted pipelining property" If TRUE, then the scheduled entities must implement pipelining. If FALSE or undefined, no pipelining applies.
response	TimingDescriptionEvent	0..1	ref	The response event representing the end point time (final `TimingDescriptionEvent`) of a timing event chain. **Tags:** atp.Splitkey=response xml.sequenceOffset=20
segment	TimingDescriptionEvent Chain	*	ref	A composed event chain consists of an arbitrary number of sub-chains. **Tags:** atp.Splitkey=segment xml.sequenceOffset=30
stimulus	TimingDescriptionEvent	0..1	ref	The stimulus event representing the starting point (initial `TimingDescriptionEvent`) of a timing event chain. **Tags:** atp.Splitkey=stimulus xml.sequenceOffset=10

Table A.1021: TimingDescriptionEventChain

Class	**TimingEvent**
Note	This event is used to start `RunnableEntity`s that shall be executed periodically.
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event

▽

△

Class	**TimingEvent**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
offset	TimeValue	0..1	attr	The value makes an assumption about the time offset of the first activation of the `RunnableEntity` triggered by the mapped `TimingEvent` relative to the periodic activation of the time base of this `TimingEvent`. Unit: second.
period	TimeValue	0..1	attr	Period of timing event in seconds. The value of this attribute shall be greater than zero.

Table A.1022: TimingEvent

Class	**TlsCryptoCipherSuite**			
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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TlsCryptoServiceMapping.tlsCipherSuite, TlsSecureComProps.tlsCipherSuite			
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.Splitkey=authentication
authentication Key	CryptoServiceKey	0..1	ref	This reference identifies the private key or pre-shared key used for authentication. **Tags:** atp.Splitkey=authenticationKey
certificate	CryptoService Certificate	0..1	ref	This reference identifies the applicable local certificate. **Tags:** atp.Splitkey=certificate
cipherSuiteId	PositiveInteger	0..1	attr	Identification of the CipherSuite according to the IANA assignments list.
cipherSuite ShortLabel	String	0..1	attr	Name of the CipherSuite according to the IANA assignments list.
ellipticCurve	CryptoEllipticCurve Props	*	ref	This references point to the properties of elliptic curves. **Tags:** atp.Splitkey=ellipticCurve
encryption	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the execution of encryption. **Tags:** atp.Splitkey=encryption
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.Splitkey=keyExchange
keyExchange Authentication	CryptoServicePrimitive	*	ref	This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm. **Tags:** atp.Splitkey=keyExchangeAuthentication
priority	PositiveInteger	0..1	attr	This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.
props	TlsCryptoCipherSuite Props	0..1	aggr	The aggregated TlsCryptoCipherSuiteProps provide details for the TLS Cipher Suite. **Tags:** atp.Splitkey=props, props.shortName

▽

△

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.Splitkey=pskIdentity
remote Certificate	CryptoService Certificate	0..1	ref	This reference identifies the applicable remote certificate. **Tags:** atp.Splitkey=remoteCertificate
signature Scheme	CryptoSignature Scheme	*	ref	This reference points to the properties of a TLS Signature Scheme. **Tags:** atp.Splitkey=signatureScheme
version	TlsVersionEnum	0..1	attr	This attribute supports the definition of the applicable version of TLS.

Table A.1023: TlsCryptoCipherSuite

Class	**TlsCryptoServiceMapping**			
Note	This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS).			
Base	*ARObject*, *CryptoServiceMapping*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cryptoServiceMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
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.Splitkey=keyExchange
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	This aggregation represents the collection of supported cipher suites. **Tags:** atp.Splitkey=tlsCipherSuite, tlsCipherSuite.shortName
useClient Authentication Request	Boolean	0..1	attr	Defines if client authentication shall be applied for this TLS connection.
useSecurity Extension RecordSize Limit	Boolean	0..1	attr	Defines if the security extension for max_fragment_length shall be supported as defined in IETF RFC 8449, chapter 4.1.

Table A.1024: TlsCryptoServiceMapping

Class	**TlsPskIdentity**			
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.			
Base	*ARObject*			
Aggregated by	TlsCryptoCipherSuite.pskIdentity			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
pskIdentity	String	0..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 A.1025: TlsPskIdentity

Class	TlvDataIdDefinition			
Note	This meta-class represents the ability to define the tlvDataId.			
Base	*ARObject*			
Aggregated by	TlvDataIdDefinitionSet.tlvDataIdDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	0..1	attr	This attribute represents the definition of the value of the TlvDataId **Stereotypes:** atpIdentityContributor
tlvArgument	ArgumentDataPrototype	0..1	ref	This reference assigns a tlvDataId to a given argument of a ClientServerOperation. **Tags:** atp.Splitkey=tlvArgument
tlv Implementation DataType Element	AbstractImplementation DataTypeElement	0..1	ref	This reference associates the definition of a TLV data id with a given AbstractImplementationDataTypeElement. **Tags:** atp.Splitkey=tlvImplementationDataTypeElement
tlvRecord Element	ApplicationRecord Element	0..1	ref	This reference associates the definition of a TLV data id with a given ApplicationRecordElement. **Tags:** atp.Splitkey=tlvRecordElement

Table A.1026: TlvDataIdDefinition

Class	TlvDataIdDefinitionSet			
Note	This meta-class acts as a container of TlvDataIdDefinitions to be used in a given context **Tags:** atp.recommendedPackage=TlvDataDefinitionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tlvDataId Definition	TlvDataIdDefinition	*	aggr	This aggregation represents the collection of TlVDataTid Definitions aggregated by the TlvDataIdDefinitionSet **Tags:** atp.Splitkey=tlvDataIdDefinition, tlvDataId Definition.id

Table A.1027: TlvDataIdDefinitionSet

Class	Topic1			
Note	This meta-class represents a topic of a documentation. Topics are similar to chapters but they cannot be nested. They also do not appear in the table of content. Topics can be used to produce intermediate headlines thus structuring a chapter internally.			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*			
Aggregated by	MsrQueryResultTopic1.topic1, TopicOrMsrQuery.topic1			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
helpEntry	String	0..1	attr	This specifies an entry point in an online help system to be linked with the parent class. The syntax shall be defined by the applied help system respectively help system generator. **Tags:** xml.attribute=true

△

Class	**Topic1**			
topicContent	TopicContentOrMsrQuery	0..1	aggr	This is the content of the topic. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.1028: Topic1

Class	**TpAddress**			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpAddress, J1939TpConfig.tpAddress, LinTpConfig.tpAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpAddress	Integer	0..1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.

Table A.1029: TpAddress

Class	***TpConfig*** (abstract)			
Note	Contains all configuration elements for AUTOSAR TP.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	CanTpConfig, DoIpTpConfig, EthTpConfig, FlexrayTpConfig, IEEE1722TpConfig, J1939TpConfig, LinTp Config, SomeipTpConfig			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Cluster	CommunicationCluster	0..1	ref	A TpConfig is existing always in the context of exactly one CommunicationCluster. **Tags:** atp.Splitkey=communicationCluster

Table A.1030: TpConfig

Class	***TpConnection*** (abstract)			
Note	TpConnection Base Class.			
Base	*ARObject*			
Subclasses	CanTpConnection, DoIpTpConnection, EthTpConnection, FlexrayTpConnection, J1939TpConnection, LinTpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ident	TpConnectionIdent	0..1	aggr	This adds the ability to become referrable to Tp Connection. **Stereotypes:** atpIdentityContributor

Table A.1031: TpConnection

Class	**TpConnectionIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable Tp Connection.			
Base	*ARObject*, *Referrable*			
Aggregated by	*TpConnection*.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.1032: TpConnectionIdent

Class	**TpPort**			
Note	Dynamic or direct assignment of a PortNumber.			
Base	*ARObject*			
Aggregated by	TcpTp.tcpTpPort, UdpTp.udpTpPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
portNumber	PositiveInteger	0..1	attr	Port Number.

Table A.1033: TpPort

Class	***Traceable*** (abstract)			
Note	This meta class represents the ability to be subject to tracing within an AUTOSAR model. Note that it is expected that its subclasses inherit either from MultilanguageReferrable or from Identifiable. Nevertheless it also inherits from MultilanguageReferrable in order to provide a common reference target for all Traceables.			
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*			
Subclasses	StructuredReq, *TimingConstraint*, TraceableTable, TraceableText			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
trace	Traceable	*	ref	This association represents the ability to trace to upstream requirements / constraints. This supports for example the bottom up tracing ProjectObjectives <- MainRequirements <- Features <- RequirementSpecs <- BSW/AI **Tags:** atp.Splitkey=trace xml.sequenceOffset=20

Table A.1034: Traceable

Class	**TraceableText**			
Note	Represents a paragraph level text which can be referenced in order to establish tracing. It supports specific tracing of document items as specified in [TPS_STDT_00098]. The following approach applies: • `shortName`: represents the tag for tracing • `longName`: represents the headline • `category`: represents the kind of the tagged text			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*, *Traceable*			
Aggregated by	DocumentationBlock.trace			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**TraceableText**			
text	DocumentationBlock	0..1	aggr	This represents the text to which the tag applies. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.1035: TraceableText

Enumeration	**TransferPropertyEnum**
Note	Transfer Properties of a Signal.
Aggregated by	ISignalToIPduMapping.transferProperty
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 A.1036: TransferPropertyEnum

Class	***TransformationDescription*** (abstract)			
Note	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.			
Base	*ARObject*, *Describable*			
Subclasses	DdsTransformationDescription, EndToEndTransformationDescription, SOMEIPTransformation Description, UserDefinedTransformationDescription			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.1037: TransformationDescription

Class	«atpVariation» ***TransformationISignalProps*** (abstract)			
Note	TransformationISignalProps holds all the attributes for the different TransformationTechnologies that are ISignal specific. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARObject*, *Describable*			
Subclasses	DdsTransformationISignalProps, EndToEndTransformationISignalProps, SOMEIPTransformationISignalProps, UserDefinedTransformationISignalProps			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
csErrorReaction	CSTransformerError ReactionEnum	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 TransfromationProps on the level of DataPrototypes. **Tags:** atp.Splitkey=dataPrototypeTransformationProps, dataPrototypeTransformationProps.ident.shortName
ident	TransformationISignal PropsIdent	0..1	aggr	This adds the ability to add a shortName to TransformationISignalProps. Please note that the short-name needs to be provided if the splitable mechanism is used. **Stereotypes:** atpIdentityContributor
transformer	Transformation Technology	0..1	ref	Reference to the TransformationTechnology description that contains transformer specific and ISignal independent configuration properties. **Tags:** atp.Splitkey=transformer

Table A.1038: TransformationISignalProps

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

Table A.1039: TransformationTechnology

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

Table A.1040: TransformerClassEnum

Class	**TransformerHardErrorEvent**			
Note	This event is raised when data are received which should trigger a Client/Server operation or an external Trigger but during transformation of the data a hard transformer error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation for which the transformer can raise this TransformerHardErrorEvent. **Tags:** atp.Splitkey=operation, operation.contextPPort, operation.targetProvidedOperation **InstanceRef implemented by:** POperationInAtomicSwcInstanceRef
requiredTrigger	Trigger	0..1	iref	This represents the Trigger for which the transformer can raise this TransformerHardErrorEvent. **Tags:** atp.Splitkey=requiredTrigger, requiredTrigger.contextRPort, requiredTrigger.targetTrigger **InstanceRef implemented by:** RTriggerInAtomicSwcInstanceRef

Table A.1041: TransformerHardErrorEvent

Class	**TransmissionAcknowledgementRequest**			
Note	Requests transmission acknowledgement that data has been sent successfully. Success/failure is reported via a SendPoint of a `RunnableEntity`.			
Base	*ARObject*			
Aggregated by	*SenderComSpec*.transmissionAcknowledge			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.1042: TransmissionAcknowledgementRequest

Class	**TransmissionComSpecProps**			
Note	This meta-class defines a set of transmission attributes which the application software is assumed to implement.			
Base	*ARObject*			
Aggregated by	*SenderComSpec*.transmissionProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataUpdate Period	TimeValue	0..1	attr	This attribute defines the period in which the application is assumed to transmit the respective data.
minimumSend Interval	TimeValue	0..1	attr	This attribute defines the minimum interval between two consecutive transmissions of the respective data the application is assumed to ensure.
onChangeData Prototype	DataPrototype Reference	*	aggr	This reference defines which DataPrototypes trigger the onChange transmission of the data. **Tags:** atp.Splitkey=onChangeDataPrototype
transmission Mode	TransmissionMode DefinitionEnum	0..1	attr	The attribute defines the mode in which the application is assumed to transmit the respective data.

Table A.1043: TransmissionComSpecProps

Class	**TransmissionModeCondition**			
Note	Possibility to attach a condition to each signal within an I-PDU. 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.			
Base	*ARObject*			
Aggregated by	TransmissionModeDeclaration.transmissionModeCondition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	0..1	aggr	Possibilities to define conditions **Tags:** atp.Splitkey=dataFilter
iSignalInIPdu	ISignalToIPduMapping	0..1	ref	Reference to a signal to which a condition is attached. **Tags:** atp.Splitkey=iSignalInIPdu

Table A.1044: TransmissionModeCondition

Class	**TransmissionModeDeclaration**			
Note	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES (True and False) for each I-PDU. 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 modeDrivenTrue Condition 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. The mixing of Transmission Mode Switch via API and signal value is not allowed.			
Base	*ARObject*			
Aggregated by	IPduTiming.transmissionModeDeclaration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
modeDriven FalseCondition	ModeDriven TransmissionMode Condition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDriven FalseConditions evaluate to true (AND associated) the transmissionModeFalseTiming shall be activated. mode DrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time. **Tags:** atp.Splitkey=modeDrivenFalseCondition

▽

△

Class	**TransmissionModeDeclaration**			
modeDriven TrueCondition	ModeDriven TransmissionMode Condition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDriven TrueConditions evaluate to true (AND associated) the transmissionModeTrueTiming shall be activated. mode DrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time. **Tags:** atp.Splitkey=modeDrivenTrueCondition
transmission ModeCondition	TransmissionMode Condition	*	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 "TransmissionMode Condition" and its sub-structure. In case the transmission mode shall be switched using the COM-API "Com_Switch IpduTxMode" there is no need for the "TransmissionMode Condition" and its sub-structure. **Tags:** atp.Splitkey=transmissionModeCondition
transmission ModeFalse Timing	TransmissionMode Timing	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. **Tags:** atp.Splitkey=transmissionModeFalseTiming
transmission ModeTrue Timing	TransmissionMode Timing	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. **Tags:** atp.Splitkey=transmissionModeTrueTiming

Table A.1045: TransmissionModeDeclaration

Class	**TransmissionModeTiming**			
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: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)			
Base	*ARObject*			
Aggregated by	TransmissionModeDeclaration.transmissionModeFalseTiming, TransmissionMode Declaration.transmissionModeTrueTiming			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cyclicTiming	CyclicTiming	0..1	aggr	Periodic Transmission Mode. **Tags:** atp.Splitkey=cyclicTiming
eventControlled Timing	EventControlledTiming	0..1	aggr	Direct Transmission Mode. **Tags:** atp.Splitkey=eventControlledTiming

Table A.1046: TransmissionModeTiming

Class	**Trigger**
Note	A trigger which is provided (i.e. released) or required (i.e. used to activate something) in the given context.
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	Trigger			
Aggregated by	AtpClassifier.atpFeature, BswModuleDescription.releasedTrigger, BswModuleDescription.requiredTrigger, ServiceInterface.trigger, TriggerInterface.trigger			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers. This Attribute is only used by the AUTOSAR Classic Platform.
triggerPeriod	MultidimensionalTime	0..1	aggr	Optional definition of a period in case of a periodically (time or angle) driven external trigger. **Tags:** atp.Splitkey=triggerPeriod This Attribute is only used by the AUTOSAR Classic Platform.

Table A.1047: Trigger

Class	TriggerIPduSendCondition			
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_TriggerIPDUSend API call.			
Base	ARObject			
Aggregated by	PduTriggering.triggerIPduSendCondition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode Declaration	ModeDeclaration	*	ref	Reference to one modeDeclaration which is OR associated in the context of the TriggerIPduSendCondition. **Tags:** atp.Splitkey=modeDeclaration

Table A.1048: TriggerIPduSendCondition

Class	TriggerInterface			
Note	A trigger interface declares a number of triggers that can be sent by an trigger source. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
trigger	Trigger	*	aggr	The Trigger of this trigger interface. **Tags:** atp.Splitkey=trigger, trigger.shortName

Table A.1049: TriggerInterface

Class	TriggerInterfaceMapping			
Note	Defines the mapping of unequal named `Trigger`s in context of two different `TriggerInterface`s.			
Base	ARObject, AtpBlueprint, Identifiable, MultilanguageReferrable, PortInterfaceMapping, Referrable			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
triggerMapping	TriggerMapping	*	aggr	Mapping of two Trigger in two different TriggerInterface **Tags:** atp.Splitkey=triggerMapping

Table A.1050: TriggerInterfaceMapping

Class	**TriggerMapping**			
Note	Defines the mapping of two particular unequally named Triggers in the given context.			
Base	*ARObject*			
Aggregated by	TriggerInterfaceMapping.triggerMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstTrigger	Trigger	0..1	ref	A Trigger to be mapped. **Tags:** atp.Splitkey=firstTrigger
secondTrigger	Trigger	0..1	ref	A Trigger to be mapped. **Tags:** atp.Splitkey=secondTrigger

Table A.1051: TriggerMapping

Class	**TriggerPortAnnotation**			
Note	Annotation to a port used for calibration regarding a certain Trigger.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.triggerPortAnnotation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
trigger	Trigger	0..1	ref	The instance of annotated trigger. **Tags:** atp.Splitkey=trigger

Table A.1052: TriggerPortAnnotation

Class	**TriggerToSignalMapping**			
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*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
systemSignal	SystemSignal	0..1	ref	This is the SystemSignal taken to transport the Trigger over the network. **Tags:** atp.Splitkey=systemSignal xml.sequenceOffset=20
trigger	Trigger	0..1	iref	This represents the Trigger that shall be used to trigger RunnableEntities deployed to a remote ECU. **Tags:** atp.Splitkey=trigger, trigger.contextComposition, trigger.contextComponent, trigger.contextPort, trigger.targetTrigger **InstanceRef implemented by:** TriggerInSystemInstanceRef

Table A.1053: TriggerToSignalMapping

Class	**UdpNmCluster**			
Note	Udp specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**UdpNmCluster**			
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.
nmImmediate NmCycleTime	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 nmImmediate NmTransmissions is greater one.
nmImmediate Nm Transmissions	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.
nmMessage TimeoutTime	TimeValue	0..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. This Attribute is only used by the AUTOSAR Classic Platform.
nmMsgCycle Time	TimeValue	0..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.
nmNetwork Timeout	TimeValue	0..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. If this attribute is not configured, the Node Identification is not used.
nmPnHandle MultipleNetwork Requests	Boolean	0..1	attr	Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).
nmRemote SleepIndication Time	TimeValue	0..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. This Attribute is only used by the AUTOSAR Classic Platform.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	0..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	EthernetPhysical Channel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to. **Tags:** atp.Splitkey=vlan

Table A.1054: UdpNmCluster

Class	**UdpNmNode**			
Note	Udp specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**UdpNmNode**			
allNmMessages KeepAwake	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
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.

Table A.1055: UdpNmNode

Class	**UdpProps**			
Note	This meta-class specifies the configuration options for UDP (User Datagram Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpProps.udpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
udpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing UDP packets.

Table A.1056: UdpProps

Class	**UdpTp**			
Note	Content Model for UDP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
udpTpPort	TpPort	0..1	aggr	Udp Port configuration. **Tags:** atp.Splitkey=udpTpPort

Table A.1057: UdpTp

Class	**UnassignFrameId**			
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. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
unassigned FrameTriggering	LinFrameTriggering	0..1	ref	The frame whose identifier is reset by this assignment. **Tags:** atp.Splitkey=unassignedFrameTriggering

Table A.1058: UnassignFrameId

Class	**Unit**			
Note	This is a physical measurement unit. All units that might be defined should stem from SI units. In order to convert one unit into another factor and offset are defined. For the calculation from SI-unit to the defined unit the factor (factorSiToUnit) and the offset (offsetSiTo Unit) are applied as follows: x [{unit}] := y * {siUnit} * factorSiToUnit [{unit}/{siUnit}] + offsetSiToUnit [{unit}] For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit) and the negation of the offset (offsetSiToUnit) are applied. y [{siUnit}] := (x*{unit} - offsetSiToUnit [{unit}]) / factorSiToUnit [{unit}/{siUnit}] **Tags:** atp.recommendedPackage=Units			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
displayName	SingleLanguageUnit Names	0..1	aggr	This specifies how the unit shall be displayed in documents or in user interfaces of tools.The displayName corresponds to the Unit.Display in an ASAM MCD-2MC file. **Tags:** atp.Splitkey=displayName xml.sequenceOffset=20
factorSiToUnit	Float	0..1	attr	This is the factor for the conversion from SI Units to units. The inverse is used for conversion from units to SI Units. **Tags:** xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits. **Tags:** xml.sequenceOffset=40
physical Dimension	PhysicalDimension	0..1	ref	This association represents the physical dimension to which the unit belongs to. Note that only values with units of the same physical dimensions might be converted. **Tags:** atp.Splitkey=physicalDimension xml.sequenceOffset=50

Table A.1059: Unit

Primitive	**UnlimitedInteger**
Note	An instance of UnlimitedInteger is an element in the set of integer numbers (..., -2, -1, 0, 1, 2, ...). The range is limited by constraint 2534. The value can be expressed in decimal, octal, hexadecimal and binary representation. Negative numbers can only be expressed in decimal notation. **Tags:** xml.xsd.customType=UNLIMITED-INTEGER xml.xsd.pattern=0|[\+\-]?[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[bB][0-1]+|0[0-7]+ xml.xsd.type=string

Table A.1060: UnlimitedInteger

Class	**UserDefinedIPdu**
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

▽

△

Class	UserDefinedIPdu			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
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 A.1061: UserDefinedIPdu

Class	UserDefinedPdu			
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	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
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 A.1062: UserDefinedPdu

Class	UserDefinedPhysicalChannel			
Note	This element allows the modeling of arbitrary Physical Channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.1063: UserDefinedPhysicalChannel

Class	«atpMixed» **ValueList**			
Note	This is a generic list of numerical values.			
Base	*ARObject*			
Aggregated by	RuleBasedAxisCont.swArraysize, RuleBasedValueCont.swArraysize, SwAxisCont.swArraysize, SwServiceArg.swArraysize, SwValueCont.swArraysize			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
v	Numerical	0..1	attr	This is a particular numerical value without variation. **Tags:** xml.sequenceOffset=30

▽

△

Class	«atpMixed» **ValueList**			
vf (ordered)	Numerical	*	attr	This is one entry in the list of numerical values **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.typeElement=false xml.typeWrapperElement=false

Table A.1064: ValueList

Class	***ValueSpecification*** (abstract)			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	*ARObject*			
Subclasses	*AbstractRuleBasedValueSpecification*, ApplicationValueSpecification, *CompositeValueSpecification*, ConstantReference, NotAvailableValueSpecification, NumericalValueSpecification, ReferenceValue Specification, TextValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
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 A.1065: ValueSpecification

Class	**VariableAccess**			
Note	The presence of a `VariableAccess` implies that a `RunnableEntity` needs access to a `VariableDataPrototype`. The kind of access is specified by the role in which the class is used.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *ReceiverComSpec*.replaceWith, RunnableEntity.dataReadAccess, Runnable Entity.dataReceivePointByArgument, RunnableEntity.dataReceivePointByValue, RunnableEntity.data SendPoint, RunnableEntity.dataWriteAccess, RunnableEntity.readLocalVariable, RunnableEntity.written LocalVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessed Variable	AutosarVariableRef	0..1	aggr	This denotes the accessed variable. **Tags:** atp.Splitkey=accessedVariable

▽

△

Class	**VariableAccess**			
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 A.1066: VariableAccess

Class	**VariableAndParameterInterfaceMapping**			
Note	Defines the mapping of `VariableDataPrototype`s or `ParameterDataPrototype`s in context of two different `SenderReceiverInterface`s, `NvDataInterface`s or `ParameterInterface`s.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataMapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular VariableData Prototypes or ParameterDataPrototypes with unequal names and/or unequal semantic (resolution or range) in context of two different SenderReceiverInterfaces, Nv DataInterfaces or ParameterInterfaces **Tags:** atp.Splitkey=dataMapping

Table A.1067: VariableAndParameterInterfaceMapping

Class	**VariableDataPrototype**			
Note	A `VariableDataPrototype` represents a formalized generic piece of information that is typically mutable by the application software layer. `VariableDataPrototype` is used in various contexts and the specific context gives the otherwise generic `VariableDataPrototype` a dedicated semantics.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.indication, *AtpClassifier*.atpFeature, BswInternalBehavior.arTypedPerInstanceMemory, BswModuleDescription.providedData, BswModuleDescription.requiredData, BulkNvDataDescriptor.bulkNvBlock, DiagnosticSovdAccessArgument.contentObject, *InternalBehavior*.staticMemory, NvBlockDescriptor.ramBlock, NvDataInterface.nvData, SenderReceiverInterface.dataElement, ServiceInterface.event, SwcInternalBehavior.arTypedPerInstanceMemory, SwcInternalBehavior.explicitInterRunnableVariable, SwcInternalBehavior.implicitInterRunnableVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype **Tags:** atp.Splitkey=initValue

Table A.1068: VariableDataPrototype

Class	**VariableDataPrototypeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype, PortElementToCommunicationResourceMapping.variableDataPrototype, QueuedReceiverComSpecProps.dataElement, SenderReceiverToSignalGroupMapping.dataElement, SenderReceiverToSignalMapping.dataElement, SignalServiceTranslationEventProps.translationTarget, SwcToSwcSignal.dataElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived

▽

△

Class	**VariableDataPrototypeInSystemInstanceRef**			
context Component (ordered)	SwComponent Prototype	*	ref	
context Composition	RootSwComposition Prototype	0..1	ref	
contextPort	PortPrototype	0..1	ref	
targetData Prototype	VariableDataPrototype	0..1	ref	

Table A.1069: VariableDataPrototypeInSystemInstanceRef

Class	**VariableInAtomicSWCTypeInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	AutosarVariableRef.autosarVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	AtomicSwComponent Type	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This is the context in a compositeDataType. **Tags:** xml.sequenceOffset=40
portPrototype	PortPrototype	0..1	ref	This is the port providing the parameter or the entry point to the parameter structure. **Tags:** xml.sequenceOffset=20
rootVariable DataPrototype	VariableDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=30
targetData Prototype	DataPrototype	0..1	ref	This is the target of the instance ref. Note that it shall be one of ApplicationCompositeElementDataPrototype of VariableDataPrototype. **Tags:** xml.sequenceOffset=50

Table A.1070: VariableInAtomicSWCTypeInstanceRef

Class	**VariationPoint**			
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***
blueprint Condition	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	MultiLanguageOverview Paragraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:** xml.sequenceOffset=20

▽

△

Class	**VariationPoint**			
formalBlueprint Generator	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 ARMQL. Note that variationPoints are not allowed within a formal BlueprintGenerator. **Tags:** atp.Status=draft xml.sequenceOffset=30
postBuildVariant Condition	PostBuildVariant Condition	*	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 splitable 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. **Stereotypes:** atpIdentityContributor **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 A.1071: VariationPoint

Class	**VariationPointProxy**			
Note	The VariationPointProxy represents variation points of the C/C++ implementation. In case of bindingTime = compileTime the RTE provides defines which can be used for Pre Processor directives to implement compileTime variability.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.variationPointProxy, SwcInternalBehavior.variationPointProxy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
conditionAccess	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. **Tags:** atp.Splitkey=conditionAccess
implementation DataType	AbstractImplementation DataType	0..1	ref	This association to ImplementationDataType shall be taken as an implementation hint by the RTE generator. **Tags:** atp.Splitkey=implementationDataType

▽

△

Class	**VariationPointProxy**			
postBuildValue Access	PostBuildVariant Criterion	0..1	ref	This represents the applicable PostBuildVariantCriterion in the context of a VariationPointProxy. Note that the technical details how to access the particular postBuildValueAccess are still considered internal to the RTE and are consequently not standardized. **Tags:** atp.Splitkey=postBuildValueAccess
postBuildVariant Condition	PostBuildVariant Condition	*	aggr	This represents that applicable PostBuoldVariant Condition in the context of aVariationPointProxy. **Tags:** atp.Splitkey=postBuildVariantCondition
valueAccess	AttributeValueVariation Point	0..1	aggr	This value acts as Binding Function for the VariationPoint. **Tags:** atp.Splitkey=valueAccess

Table A.1072: VariationPointProxy

Class	**VfbTiming**			
Note	A model element used to define timing descriptions and constraints at VFB level. TimingDescriptions aggregated by VfbTiming are restricted to event chains referring to events which are derived from the class TDEventVfb. **Tags:** atp.recommendedPackage=TimingExtensions			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
component	SwComponentType	0..1	ref	This defines the scope of a VfbTiming. All corresponding timing descriptions and constraints shall be defined within this scope. **Tags:** atp.Splitkey=component

Table A.1073: VfbTiming

Class	**VlanConfig**			
Note	VLAN Configuration attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetPhysicalChannel.vlan			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
vlanIdentifier	PositiveInteger	0..1	attr	A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.

Table A.1074: VlanConfig

Class	**VlanMembership**
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*
Aggregated by	CouplingPort.vlanMembership

▽

△

Class	**VlanMembership**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultPriority	PositiveInteger	0..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. **Tags:** atp.Splitkey=dhcpAddressAssignment
sendActivity	EthernetSwitchVlan EgressTaggingEnum	0..1	attr	Attribute denotes whether a VLAN tagged ethernet frame will be 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	EthernetPhysical Channel	0..1	ref	References a channel that represents a VLAN or an untagged channel. **Tags:** atp.Splitkey=vlan

Table A.1075: VlanMembership

Class	**WaitPoint**			
Note	This defines a wait-point for which the `RunnableEntity` can wait.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RunnableEntity.waitPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
timeout	TimeValue	0..1	attr	Time in seconds before the WaitPoint times out and the blocking wait call returns with an error indicating the timeout.
trigger	RTEEvent	0..1	ref	This is the `RTEEvent` this `WaitPoint` is waiting for. **Tags:** atp.Splitkey=trigger

Table A.1076: WaitPoint

Class	**WorstCaseHeapUsage**			
Note	Provides a formal worst case heap usage.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
memory Consumption	PositiveInteger	0..1	attr	Worst case heap consumption. Unit: byte.

Table A.1077: WorstCaseHeapUsage

Class	**WorstCaseStackUsage**			
Note	Provides a formal worst case stack usage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
memory Consumption	PositiveInteger	0..1	attr	Worst case stack consumption. Unit: byte.

Table A.1078: WorstCaseStackUsage

Class	**Xdoc**			
Note	This meta-class represents the ability to refer to an external document which can be rendered as printed matter.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForParagraph*.xdoc			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
date	DateTime	0..1	attr	This element specifies the release date of the external document if applicable. **Tags:** xml.sequenceOffset=50
number	String	0..1	attr	This represents document number of an external document that is referenced. Kept as a string. **Tags:** xml.sequenceOffset=30
position	String	0..1	attr	This represents the reference to the relevant positions of a standard. Kept as a string. **Tags:** xml.sequenceOffset=80
publisher	String	0..1	attr	This represents the publisher of an external document that is being referenced. Kept as a string. **Tags:** xml.sequenceOffset=60
state	String	0..1	attr	This represents version and state of the external document. Kept as a string. **Tags:** xml.sequenceOffset=40
url	Url	0..1	aggr	This specifies the URL of the external document. **Tags:** atp.Splitkey=url xml.sequenceOffset=70

Table A.1079: Xdoc

Class	**Xfile**			
Note	This represents to reference an external file within a documentation.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForParagraph*.xfile			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tool	String	0..1	attr	This element describes the tool which was used to generate the corresponding Xfile . Kept as a string since no specific syntax can be provided to denote a tool. **Tags:** xml.sequenceOffset=50



△

Class	**Xfile**			
toolVersion	String	0..1	attr	This element describes the tool version which was used to generate the corresponding xfile. Kept as a string, since no specific syntax can be specified. **Tags:** xml.sequenceOffset=60
url	Url	0..1	aggr	This represents the URL of the external file. **Tags:** atp.Splitkey=url xml.sequenceOffset=30

Table A.1080: Xfile

Class	**XrefTarget**			
Note	This element specifies a reference target which can be scattered throughout the text.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForOverviewParagraph*.xrefTarget, *MixedContentForParagraph*.xrefTarget			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.1081: XrefTarget

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R26-11

B.1.1 Added Constraints in R26-11

Number	Heading
[constr_10744]	Multiplicity of the reference in the role `DataPrototypeMapping.textTableMapping`
[constr_10745]	Multiplicity of the reference in the role `SubElementMapping.textTableMapping`
[constr_10750]	Existence of `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping`
[constr_10751]	Existence of reference to `EventObdReadinessGroup`
[constr_10752]	Existence of `DiagnosticReadDataByPeriodicID`
[constr_10792]	Relation between `baseTypeSize`, `baseTypeEncoding` and `InternalConstrs`
[constr_10793]	Restriction for the definition of `internalConstrs` in the context of an `ImplementationDataType` of `category TYPE_REFERENCE`
[constr_2542]	Value of `AttributeValueVariationPoint.bindingTime`
[constr_2643]	`SwSystemconst` reference assignment to `SwSystemconstValue`
[constr_2644]	Assign the attribute `SwSystemconstValue.value` to `SwSystemconstValue`
[constr_3826]	`DdsCpTopic.ddsPartition` reference restriction
[constr_3827]	Existence of `IkePqPresharedKey.pqPresharedKeyId`
[constr_3828]	Existence of `IkePqPresharedKey.pqPresharedKey`
[constr_3829]	`CommunicationCluster.comMLightTimeout` not applicable for `EthernetCluster`
[constr_4200]	Existence of `EcucDefinitionCollection.module`
[constr_4201]	Existence of `EcucModuleDef.container`
[constr_4202]	Existence of `EcucValueCollection.ecucValue`
[constr_4203]	`EcucModuleConfigurationValues.container`
[constr_6922]	Restriction of `System.category` in a `SystemTiming` view
[constr_6923]	Restriction of `System.category` in a `VfbTiming` view
[constr_6924]	Multiplicity of `TDEventOccurrenceExpression.function`
[constr_6925]	Existence of `PortPrototypeBlueprint.interface`
[constr_6926]	Restriction of `successor` and `directSuccessor`
[constr_6927]	Exclusivity of either `EOCExecutableEntityRef` or `EOCEventRef` in an `ExecutionOrderConstraint`

▽

△

Number	Heading
[constr_6928]	Exclusivity of component and bswModuleInstance in an EOCExecutableEntityRef
[constr_6929]	Exclusivity of component and bswModuleInstance in an EOCEventRef
[constr_6930]	Restriction of ExecutionTimeConstraint.component
[constr_6931]	Affinity of BswModuleEntity in GET/RT
[constr_6932]	Affinity of RunnableEntity in GET/RT
[constr_6933]	BlueprintMapping.derivedObject shall not refer to a subclass of AtpBlueprint
[constr_9398]	dataTypePolicy = transformingISignal not applicable for ISignals that are transformed by the ComBasedTransformer
[constr_9399]	Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if no data transformation is used
[constr_9400]	Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if comBasedSignalGroup Transformation is used
[constr_9401]	Mapping of VariableDataPrototype typed by a array data type consisting of UINT8 elements to exactly one SystemSignal
[constr_9402]	PduActivationRoutingGroup with PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast that is aggregated by a ConsumedEventGroup
[constr_9403]	J1939TpProtocolTypeEnum values fd_bam, fd_bam_cmdt, and fd_cmdt shall only be used on CAN FD clusters
[constr_9404]	J1939Cluster.usesMultiPg attribute shall only be set to a value on CAN FD clusters
[constr_9407]	Standardized values for the attribute category of meta-class PortElementToCommunicationResourceMapping
[constr_9408]	J1939TpConnection.tpPg in case J1939TpConnection.tpProtocolType is fpp
[constr_9409]	Existence of AtpInstanceRef.atpTarget
[constr_9410]	Existence of ConditionByFormula.bindingTime
[constr_9411]	Existence of PostBuildVariantCondition.matchingCriterion
[constr_9412]	Existence of PostBuildVariantCondition.value
[constr_9413]	Existence of PostBuildVariantCriterion.compuMethod
[constr_9414]	Existence of EvaluatedVariantSet.approvalStatus
[constr_9415]	Existence of PostBuildVariantCriterionValue.variantCriterion
[constr_9416]	Existence of PostBuildVariantCriterionValue.value
[constr_9417]	Existence of AnyInstanceRef.target
[constr_9418]	Existence of ShortNameFragment.fragment
[constr_9419]	Existence of ShortNameFragment.role

▽

△

Number	Heading
[constr_9420]	Existence of `ReferenceBase.shortLabel`
[constr_9421]	Exactly two `MacSecConnectivityAssociation.couplingPort` references for a switched Ethernet connection
[constr_9422]	Consistent reference to a VLAN using `AbstractMacSecCommunication.vlan`
[constr_9423]	Coverage of all VLANs referenced by a `CouplingPort` by `MacSecConnectivityAssociation` and `MacSecBypassedCommunication`
[constr_9424]	10BASE-T1S `MacSecConnectivityAssociation` `CouplingPortConnection` reference consistency
[constr_9425]	Existence of `CanSecSecureChannel.csci`
[constr_9426]	Coverage of all `CanFrameTriggering`s and `SoConIPduIdentifier`s in the scope of a `CanCommunicationConnector` by `MessageToCanSecSecureChannelMapping` and `MessageToBypassCanSecCommunicationMapping`
[constr_9427]	Existence of `MessageToCanSecSecureChannelMapping.secureChannel`
[constr_9428]	Consistent `MessageToCanSecSecureChannelMapping.secureChannel` and `AbstractMessageToSecureChannelMapping.canFrameTriggering` mapping
[constr_9439]	Existence of attribute `EcucModuleDef.postBuildVariantSupport`
[constr_9440]	Existence of attribute `EcucModuleDef.postBuildLoadableSupport`
[constr_9441]	Value of `EcucContainerDef.postBuildLoadableMultiplicity` if `postBuildLoadableSupport` is set to `false`
[constr_9442]	Value of `postBuildLoadableMultiplicity` if `postBuildLoadableSupport` is set to `false`
[constr_9443]	Value of `postBuildLoadableValue` if `postBuildLoadableSupport` is set to `false`
[constr_9444]	Mutual exclusivity of `linkTimeSupport` and `postBuildLoadableMultiplicity`
[constr_9445]	`EcucContainerDef.linkTimeSupport` not applicable in StMD
[constr_9446]	Existence of the `postBuildLoadableMultiplicity` attribute
[constr_9447]	Removing an instance of the `EcucContainerDef` at post-build time
[constr_9450]	`EcucCommonAttributes.linkTimeSupport` not applicable in StMD
[constr_9451]	Existence of attribute `EcucCommonAttributes.linkTimeSupport`
[constr_9452]	Mutual exclusivity of `linkTimeSupport` and `postBuildLoadableMultiplicity`
[constr_9453]	Mutual exclusivity of `linkTimeSupport` and `postBuildLoadableValue`
[constr_9454]	Existence of attribute `EcucCommonAttributes.postBuildVariantValue`
[constr_9455]	Existence of attribute `EcucCommonAttributes.` `postBuildVariantMultiplicity`
[constr_9456]	Existence of attribute `EcucCommonAttributes.postBuildLoadableValue`
[constr_9457]	Existence of attribute `EcucCommonAttributes.` `postBuildLoadableMultiplicity`
[constr_9458]	`EcucModuleDef` that relies on `EcucCommonAttributes` with `postBuildLoadableValue` set to true of another `EcucModuleDef`

▽

△

Number	Heading
[constr_9459]	EcucModuleDef that relies on EcucCommonAttributes with postBuildLoadableMultiplicity set to true of another EcucModuleDef

Table B.1: Added Constraints in R26-11

B.1.2 Changed Constraints in R26-11

Number	Heading
[constr_1004]	Mapping of ApplicationDataTypes in the scope of single AtomicSwComponentTypes and ParameterSwComponentTypes
[constr_10606]	Existence of ConstantSpecificationMapping or CalibrationParameterValue for ApplicationValueSpecification or ApplicationRuleBasedValueSpecification of category COM_AXIS, STD_AXIS, RES_AXIS, CURVE, MAP, CUBOID, CUBE_4, and CUBE_5
[constr_10656]	Value of a SwSystemconst used in the context of the *formula language for post-build configuration*
[constr_1289]	Allowed Attributes vs. category for DataPrototypes typed by ApplicationDataTypes
[constr_1368]	Limitation of the target of references from DiagnosticConnection
[constr_2504]	Constraint to bindingTime
[constr_2505]	Multiplicity after binding
[constr_2507]	EvaluatedVariantSet shall not refer to itself
[constr_2512]	shortName uniqueness constraint for variants
[constr_2514]	shortLabel in VariationPoint shall be unique
[constr_2516]	Return type of an AttributeValueVariationPoint
[constr_2517]	postBuildVariantCondition only for PostBuild
[constr_2518]	Binding time is constrained
[constr_2519]	PredefinedVariants need to be consistent
[constr_2521]	The shortLabel in AttributeValueVariationPoint shall be unique
[constr_2547]	Ordered collections cannot be split into different partial models
[constr_2557]	No VariationPoints where vh.latestBindingTime set to BlueprintDerivationTime in system configurations
[constr_2558]	If vh.latestBindingTime is BlueprintDerivationTime then there shall only be blueprintCondition or formalBlueprintGenerator respectively blueprintValue
[constr_2559]	No nested VariationPoint
[constr_2567]	Undefined Value in Attribute Value Blueprints
[constr_2575]	blueprintValue in blueprints only
[constr_2577]	Binding Time in Aggregation Pattern
[constr_2578]	Binding Time in Association Pattern

▽

△

Number	Heading
[constr_2580]	Binding Time in Property Set Pattern
[constr_2587]	No System in AnyInstanceRef
[constr_2594]	Cyclic value assignments to SwSystemconst is not allowed
[constr_2599]	Maximum one VariationPoints in ≪atpMixed≫
[constr_2601]	Value of AbstractEnumerationValueVariationPoint
[constr_2626]	atpTarget of InstanceRefs shall be consistent
[constr_2634]	Ordered collections in Conditionals with property set patterns
[constr_2638]	Variation points shall not exists in non-variant roles
[constr_3025]	Usage of NPdus in TpConnections
[constr_3116]	Overlap of ClientIdRanges in the context of the enclosing System
[constr_3117]	Allowed value of attribute clientId
[constr_3120]	EcucDestinationUriPolicy.container always required
[constr_3136]	Allowed payload of SecuredIPdus
[constr_3141]	Only IPdus shall be part of a ContainerIPdu
[constr_3142]	Mandatory headerIdLongHeader for longHeader
[constr_3143]	Mandatory headerIdShortHeader for shortHeader
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m
[constr_3208]	executeDespiteDataUnavailability usage restriction for received ISignals
[constr_3325]	SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute value settings for standardized AUTOSAR security profiles
[constr_3402]	Mandatory offset if noHeader is used
[constr_3404]	Usage of ContainedIPduProps.updateIndicationBitPosition
[constr_3454]	Unique headerIdLongHeader for acceptConfigured
[constr_3455]	Unique headerIdShortHeader for acceptConfigured
[constr_3767]	CommunicationConnector.nmVariant setting to slavePassive
[constr_3768]	CommunicationConnector.nmVariant setting to slaveActive
[constr_3769]	CommunicationConnector.nmVariant setting to full
[constr_3770]	CommunicationConnector.nmVariant setting to passive
[constr_3794]	Usage of PduBufferAlignment
[constr_4504]	Restriction of the scope of an AgeConstraint
[constr_4505]	Specifying minimum and maximum number of occurrences
[constr_4506]	Specifying minimum inter-arrival time and pattern length
[constr_4507]	Specifying pattern length, pattern jitter and patter period
[constr_4508]	Multiplicity of TDEventVfbPort.portPrototypeBlueprint

▽

△

Number	Heading
[constr_4512]	Restriction of TDEventSwcInternalBehavior.variableAccess
[constr_4513]	SynchronizationTimingConstraint shall reference at least two events
[constr_4514]	SynchronizationTimingConstraint shall reference at least two event chains
[constr_4515]	Orthogonality of stimulus and response in a TimingDescriptionEventChain
[constr_4518]	Specifying end-points of a composed TimingDescriptionEventChain
[constr_4519]	Specifying patternLength
[constr_4521]	Specifying attribute synchronizationConstraintType
[constr_4522]	SynchronizationTimingConstraint shall either reference events or event chains
[constr_4523]	Restriction of maxCycleRepetitions and maxSlotsPerCycle to Repetitive Execution Order Constraint
[constr_4528]	The *root* EOCExecutableEntityRefGroup shall reference only EOCExecutableEntityRefGroups
[constr_4529]	Number of nested elements referenced by the *root* EOCExecutableEntityRefGroup
[constr_4530]	An EOCExecutableEntityRefGroup representing a cycle shall reference only EOCExecutableEntityRefs respectively EOCEventRefs
[constr_4531]	Number of nested elements referenced by EOCExecutableEntityRefGroup representing a cycle
[constr_4532]	Successor relationship is not self-referencing
[constr_4533]	Maximum number of successor relationships
[constr_4540]	maxCycleRepetitions and maxSlotsPerCycle shall not be zero
[constr_4541]	Multiplicity of EOCExecutableEntityRef.executable in an Ordinary Execution Order Constraint
[constr_4542]	Multiplicity of EOCExecutableEntityRef.executable in a Hierarchical Execution Order Constraint
[constr_4543]	Maximum value of minimumInterArrivalTime
[constr_4544]	Specifying patternLength, patternJitter and patternPeriod
[constr_4548]	Multiplicity of EOCEventRef.event in an Ordinary Execution Order Constraint
[constr_4549]	Multiplicity of EOCEventRef.event in a Hierarchical Execution Order Constraint
[constr_4554]	Restriction of the referenced TimingDescriptionEventChain for a letInterval
[constr_4561]	Usage of the category value DISPATCH_ENTRY_POINT in TimingDescriptionEvent
[constr_4562]	Usage of the category value DISPATCH_EXIT_POINT in TimingDescriptionEvent
[constr_4565]	Consistency of TDCpSoftwareClusterMapping.timingDescription and TDCpSoftwareClusterResourceMapping.timingDescription
[constr_5074]	PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast that is aggregated by an EventHandler

▽

△

Number	Heading
[constr_5075]	Allowed references of SoConIPduIdentifiers by PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast and allowed SoConIPduIdentifier references
[constr_5361]	MACsec configuration is only allowed on CouplingPorts aggregated by switch ports and EthernetCommunicationController
[constr_5362]	Relation between ContainerIPdu and contained PduTriggerings on sender side
[constr_5380]	Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of one VLAN
[constr_5506]	Existence of postBuildVariantMultiplicity attribute
[constr_5520]	Latest configuration time to change the values of symbolicNameValue parameters
[constr_5521]	Latest configuration time to change the number of instances of symbolicNameValue parameters
[constr_6816]	Restricted usage of TimingDescriptionEventChain.isPipeliningPermitted in TimingDescriptionEventChain
[constr_6817]	Restricted usage of TimingDescriptionEvent.clockReference
[constr_6818]	Multiplicity of EventTriggeringConstraint.event
[constr_6819]	Multiplicity of PeriodicEventTriggering.jitter
[constr_6820]	Multiplicity of PeriodicEventTriggering.minimumInterArrivalTime
[constr_6821]	Multiplicity of PeriodicEventTriggering.period
[constr_6822]	Multiplicity of SporadicEventTriggering.maximumInterArrivalTime
[constr_6823]	Multiplicity of SporadicEventTriggering.minimumInterArrivalTime
[constr_6824]	Multiplicity of ConcretePatternEventTriggering.patternLength
[constr_6825]	Multiplicity of BurstPatternEventTriggering.maxNumberOfOccurrences
[constr_6826]	Multiplicity of BurstPatternEventTriggering.minimumInterArrivalTime
[constr_6827]	Multiplicity of BurstPatternEventTriggering.patternLength
[constr_6828]	Multiplicity of ArbitraryEventTriggering.minimumDistance
[constr_6829]	Multiplicity of ArbitraryEventTriggering.maximumDistance
[constr_6830]	Multiplicity of ConfidenceInterval.lowerBound
[constr_6831]	Multiplicity of ConfidenceInterval.propability
[constr_6832]	Multiplicity of ConfidenceInterval.upperBound
[constr_6833]	Multiplicity of ExecutionOrderConstraint.orderedElement
[constr_6834]	Multiplicity of EOCExecutableEntityRefGroup.nestedElement
[constr_6835]	Multiplicity of ExecutionTimeConstraint.executionTimeType
[constr_6836]	Multiplicity of ExecutionTimeConstraint.executable
[constr_6837]	Multiplicity of LatencyTimingConstraint.latencyConstraintType
[constr_6838]	Multiplicity of LatencyTimingConstraint.maximum

▽

△

Number	Heading
[constr_6839]	Multiplicity of LatencyTimingConstraint.minimum
[constr_6841]	Multiplicity of LatencyTimingConstraint.scope
[constr_6842]	Multiplicity of OffsetTimingConstraint.maximum
[constr_6843]	Multiplicity of OffsetTimingConstraint.minimum
[constr_6844]	Multiplicity of OffsetTimingConstraint.source
[constr_6845]	Multiplicity of OffsetTimingConstraint.target
[constr_6846]	Multiplicity of SynchronizationTimingConstraint.synchronizationConstraintType
[constr_6847]	Multiplicity of SynchronizationTimingConstraint.tolerance
[constr_6849]	Multiplicity of SystemTiming.system
[constr_6850]	Multiplicity of BswModuleTiming.behavior
[constr_6851]	Multiplicity of BswCompositionTiming.implementation
[constr_6852]	Multiplicity of EcuTiming.ecuConfiguration
[constr_6853]	Multiplicity of ModeInBswInstanceRef.contextModeDeclarationGroupPrototype
[constr_6854]	Multiplicity of ModeInBswInstanceRef.targetModeDeclaration
[constr_6855]	Multiplicity of ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype
[constr_6856]	Multiplicity of ModeInSwcInstanceRef.contextPort
[constr_6857]	Multiplicity of ModeInSwcInstanceRef.targetModeDeclaration
[constr_6858]	Multiplicity of TDEventBswInternalBehavior.tdEventBswInternalBehaviorType
[constr_6859]	Multiplicity of TDEventBswInternalBehavior.bswModuleEntity
[constr_6860]	Multiplicity of TDEventBswModule.tdEventBswModuleType
[constr_6861]	Multiplicity of TDEventBswModule.bswModuleEntry
[constr_6862]	Multiplicity of TDEventBswModeDeclaration.tdEventBswModeDeclarationType
[constr_6863]	Multiplicity of TDEventBswModeDeclaration.modeDeclaration
[constr_6864]	Multiplicity of TDEventISignal.tdEventType
[constr_6865]	Multiplicity of TDEventISignal.iSignal
[constr_6866]	Multiplicity of TDEventISignal.physicalChannel
[constr_6867]	Multiplicity of TDEventIPdu.tdEventType
[constr_6868]	Multiplicity of TDEventIPdu.iPdu
[constr_6869]	Multiplicity of TDEventIPdu.physicalChannel
[constr_6870]	Multiplicity of TDEventFrame.tdEventType
[constr_6871]	Multiplicity of TDEventFrame.frame
[constr_6872]	Multiplicity of TDEventFrame.physicalChannel
[constr_6873]	Multiplicity of TDEventFrameEthernet.tdEventType
[constr_6874]	Multiplicity of TDHeaderIdRange.maxHeaderId

▽

△

Number	Heading
[constr_6875]	Multiplicity of TDHeaderIdRange.minHeaderId
[constr_6876]	Multiplicity of TDEventCycleStart.cycleRepetition
[constr_6877]	Multiplicity of TDEventFrClusterCycleStart.frCluster
[constr_6879]	Multiplicity of TDEventOccurrenceExpression.formula
[constr_6880]	Multiplicity of AutosarVariableInstance.variableInstance
[constr_6881]	Multiplicity of AutosarOperationArgumentInstance.operationArgumentInstance
[constr_6882]	Multiplicity of TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType
[constr_6883]	Multiplicity of TDEventSwcInternalBehavior.runnable
[constr_6884]	Multiplicity of TDEventSwcInternalBehaviorReference.referencedTDEventSwc
[constr_6885]	Multiplicity of TDEventVfbPort.isExternal
[constr_6886]	Multiplicity of TDEventVfbReference.referencedTDEventVfb
[constr_6887]	Multiplicity of TDEventVariableDataPrototype.tdEventVariableDataPrototypeType
[constr_6888]	Multiplicity of TDEventVariableDataPrototype.dataElement
[constr_6889]	Multiplicity of TDEventOperation.tdEventOperationType
[constr_6890]	Multiplicity of TDEventOperation.operation
[constr_6891]	Multiplicity of TDEventModeDeclaration.tdEventModeDeclarationType
[constr_6892]	Multiplicity of TDEventModeDeclaration.modeDeclaration
[constr_6893]	Multiplicity of TDEventTrigger.tdEventTriggerType
[constr_6894]	Multiplicity of TDEventTrigger.trigger
[constr_6895]	Multiplicity of TimingDescriptionEventChain.response
[constr_6896]	Multiplicity of TimingDescriptionEventChain.stimulus
[constr_6898]	Multiplicity of ConcretePatternEventTriggering.offset
[constr_6899]	Multiplicity of ModeInSwcInstanceRef.base
[constr_6900]	Dual existence of TDEventVfb.portPrototype and TDEventVfb.portPrototypeBlueprint
[constr_6901]	Multiplicity of TDEventBsw.bswModuleDescription
[constr_6906]	Conformity of stimulus and response in a TimingDescriptionEventChain
[constr_6908]	Restriction of EOCExecutableEntityRefGroup.letDataExchangeParadigm
[constr_6909]	Singleton ROOT_GROUP in an Execution Order Constraint
[constr_6910]	Referencing **FROM** a ROOT_GROUP in an Execution Order Constraint
[constr_6911]	Referencing **TO** a ROOT_GROUP in an Execution Order Constraint
[constr_6912]	Mandatory specification of LET interval **recurrence**
[constr_6913]	Restriction on RTEEvents used in an LET interval
[constr_6914]	Restriction of the portPrototype context of an AgeConstraint
[constr_6915]	Affinity of ISignal in TDEventISignal
[constr_6916]	Affinity of Frame in TDEventFrame

▽

△

Number	Heading
[constr_6917]	Affinity of IPdu in TDEventIPdu
[constr_6918]	Referenced TimingDescriptions in TDCpSoftwareClusterMapping and TDCpSoftwareClusterResourceMapping
[constr_6919]	Referenced CpSoftwareCluster of TDCpSoftwareClusterMapping
[constr_6920]	Multiplicity of LatencyTimingConstraint.minimum used in an LET interval
[constr_6921]	Disallow TimingDescriptionEventChain segmental circular-referencing
[constr_9357]	Existence of attributes of McDataInstance depending on the category

Table B.2: Changed Constraints in R26-11

B.1.3 Deleted Constraints in R26-11

Number	Heading
[constr_10000]	Existence of attribute RptExecutableEntityProperties.rptExecutionControl
[constr_10001]	Existence of attribute RptExecutableEntityProperties.rptServicePoint
[constr_10024]	Existence of reference in the role DiagnosticSecurityEventReportingModeMapping.dataElement
[constr_10025]	Existence of reference in the role DiagnosticSecurityEventReportingModeMapping.securityEvent
[constr_1682]	Existence of the attribute RoleBasedDataAssignment.usedDataElement.localVariable for the value RoleBasedDataAssignment.role = V2xFacVdp
[constr_1988]	Existence of attribute RptProfile.maxServicePointId
[constr_1989]	Existence of attribute RptProfile.minServicePointId
[constr_1990]	Existence of attribute RptProfile.servicePointSymbolPost
[constr_1991]	Existence of attribute RptProfile.servicePointSymbolPre
[constr_1992]	Existence of attribute RptProfile.stimEnabler
[constr_1993]	Existence of attribute RptImplPolicy.rptEnablerImplType
[constr_1994]	Existence of attribute RptImplPolicy.rptPreparationLevel
[constr_1995]	Existence of attribute RptSwPrototypingAccess.rptHookAccess
[constr_1996]	Existence of attribute RptSwPrototypingAccess.rptReadAccess
[constr_1997]	Existence of attribute RptSwPrototypingAccess.rptWriteAccess
[constr_1998]	Existence of attribute RptExecutableEntityProperties.maxRptEventId
[constr_1999]	Existence of attribute RptExecutableEntityProperties.minRptEventId
[constr_2520]	Nesting of lists shall be limited
[constr_2524]	Non splitable elements in one file
[constr_2566]	Blueprintmapping shall map appropriate elements
[constr_2579]	Binding Time in Attribute Value Pattern

▽

△

Number	Heading
[constr_2597]	ClientServerOperationBlueprintMapping constrains number of arguments
[constr_2598]	ClientServerOperationBlueprintMapping constrains the types of arguments
[constr_2602]	Completeness of AnyInstanceRef referencing ImplementationDataTypeElement
[constr_2603]	Use of appliesTo in context of the specification level
[constr_2604]	Allowed up-traces in context of appliesTo values
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3058]	References from SenderRecArrayElementMapping and from SenderRecRecordElementMapping to SystemSignals are not allowed within a SenderReceiverCompositeElementToSignalMapping
[constr_3091]	Multiplicity of valueConfigClass
[constr_3092]	Usage of configVariant and configClass attributes
[constr_3144]	Mandatory IPdu.containedIPduProps for contained IPdus
[constr_3233]	EcucModuleDef that relies on EcucCommonAttributes with valueConfigClass set to Link/PostBuild of another EcucModuleDef
[constr_3234]	EcucModuleDef that relies on EcucCommonAttributes with multiplicityConfigClass set to Link/PostBuild of another EcucModuleDef
[constr_3235]	EcucModuleDef that relies on EcucContainerDefs with multiplicityConfigClass set to Link/PostBuild of another EcucModuleDef
[constr_3236]	EcucModuleDef that relies on EcucCommonAttributes with postBuildVariantValue set to true of another EcucModuleDef
[constr_3237]	EcucModuleDef that relies on EcucCommonAttributes with postBuildVariantMultiplicity set to true of another EcucModuleDef
[constr_3238]	EcucModuleDef that relies on EcucContainerDef with postBuildVariantMultiplicity set to true of another EcucModuleDef
[constr_3249]	Category of HwElement for SwcToEcuMapping
[constr_3327]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_22
[constr_3449]	Impact of postBuildVariantUsed value set to FALSE
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3573]	EcucAbstractConfigurationClass.configClass always required
[constr_3574]	EcucAbstractConfigurationClass.configVariant always required
[constr_3583]	EcucDestinationUriPolicy.destinationUriNestingContract always required
[constr_3590]	EcucModuleConfigurationValues.implementationConfigVariant always required
[constr_4500]	Restricted usage of Occurence Expression functions
[constr_4502]	Use references only as function operands
[constr_4503]	Restricted usage of AutosarOperationArgumentInstance for Content Filter

▽

△

Number	Heading
[constr_4510]	Specifying references to RunnableEntity and VariableAccess
[constr_4511]	Validity of referencing RunnableEntity
[constr_4516]	Completeness of a composed TimingDescriptionEventChain
[constr_4520]	Specifying attribute synchronizationConstraintType
[constr_4525]	Precedence of successor relationships successor and directSuccessor
[constr_4526]	Specifying maxCycleRepetitions and maxSlotsPerCycle in a Repetitive Execution Order Constraint
[constr_4527]	Referencing TimingDescriptionEvent in a Repetitive Execution Order Constraint
[constr_4534]	Maximum number of directSuccessor relationships
[constr_4536]	Compatible recurrence of any ExecutableEntity
[constr_4537]	References among elements in an ExecutionOrderConstraint
[constr_4538]	Hierarchical Execution Order Constraint: EOCExecutableEntityRef, EOCEventRef, and EOCExecutableEntityRefGroup shall be target or source of a successor relationship
[constr_4539]	The successor relationships successor and directSuccessor shall not be used
[constr_4545]	Referring either ExecutableEntitys or AbstractEvents
[constr_4546]	Setting the attribute isEvent
[constr_4547]	Restriction of ExecutionOrderConstraint.permitMultipleReferencesToEE
[constr_4551]	Use only Numericals in TDEventOccurrenceExpression
[constr_4552]	Restricted usage of AutosarVariableInstance for Content Filter
[constr_4559]	Restriction of TimingDescriptionEvent.category
[constr_5015]	Multiplicity of multiplicityConfigClass
[constr_5269]	Exclusion of ContainedIPduProps.containedPduTriggering reference
[constr_5270]	Exclusive usage of ContainerIPdu.containedPduTriggering and ContainerIPdu.containedIPduTriggeringProps
[constr_5315]	FlexrayArTpConnections within the same FlexrayArTpChannel not allowed to have the same address information
[constr_5375]	IPdu shall only be referenced once from a FlexrayArTpConnection in the role directTpSdu or reversedTpSdu on a FlexrayCluster
[constr_5465]	Existence of softwareComposition
[constr_5475]	Existence of SenderReceiverCompositeElementToSignalMapping.systemSignal
[constr_5476]	Existence of SenderReceiverCompositeElementToSignalMapping.typeMapping
[constr_5500]	Applicability of the multiplicityConfigClass attribute
[constr_5502]	Introduction of new EcucParameterValues of type EcucFunctionNameDef at post-build time
[constr_5504]	Removing an instance of the EcucContainerDef at post-build time
[constr_5508]	Applicability of postBuildVariantMultiplicity attribute

▽

△

Number	Heading
[constr_5512]	postBuildVariantValue attribute of symbolicNameValue parameters
[constr_5514]	Applicability of the multiplicityConfigClass attribute
[constr_5522]	postBuildVariantMultiplicity attribute of symbolicNameValue parameters
[constr_5523]	Allowed configClasses for paired configVariants
[constr_6897]	Existence of TimingDescriptionEventChain.segment
[constr_6907]	Restriction of EOCExecutableEntityRefGroup.triggeringEvent
[constr_9238]	Existence of FlexrayArTpChannel.ackType
[constr_9239]	Existence of FlexrayArTpChannel.extendedAddressing
[constr_9240]	Existence of FlexrayArTpChannel.maximumMessageLength
[constr_9241]	Existence of FlexrayArTpChannel.minimumSeparationTime
[constr_9242]	Existence of FlexrayArTpChannel.multicastSegmentation
[constr_9243]	Existence of FlexrayArTpChannel.tpConnection
[constr_9244]	Existence of FlexrayArTpConnection.directTpSdu
[constr_9245]	Existence of FlexrayArTpConnection.source
[constr_9246]	Existence of FlexrayArTpConnection.target
[constr_9268]	Existence of J1939TpConnection.dataPdu
[constr_9269]	Existence of J1939TpConnection.flowControlPdu

Table B.3: Deleted Constraints in R26-11

B.2 Traceable item history of this document according to AUTOSAR Release R25-11

B.2.1 Added Constraints in R25-11

Number	Heading
[constr_10632]	Compatibility of SwTextProps
[constr_10642]	Existence of DiagnosticEventPortMapping.diagnosticEvent for one specific DiagnosticEvent
[constr_10645]	Value of DiagnosticEvent.confirmationThreshold for an n:1 mapping to DiagnosticTroubleCodeUds
[constr_10646]	Value of attributes of DiagnosticConnectedIndicator for an n:1 mapping to DiagnosticTroubleCodeUds where all affected DiagnosticEvents transitively refer to the identical DiagnosticIndicator
[constr_10647]	In an n:1 mapping between DiagnosticEvent and DiagnosticTroubleCodeUds, all DiagnosticEvents shall be mapped to the same DiagnosticOperationCycle
[constr_10656]	Value of a SwSystemconst used in the context of the *formula language for post-build configuration*

▽

△

Number	Heading
[constr_10665]	Existence of attribute DiagnosticCommonProps.authenticationTimeout
[constr_10667]	Simple compatibility of ImplementationDataTypes
[constr_10668]	Restriction for the existence of attributes of RunnableEntityArgument for port-defined argument values
[constr_10669]	Existence of reference RunnableEntityArgument.implementationDataType
[constr_10670]	Compatibility between the corresponding ClientServerOperation.argument and RunnableEntity.argument
[constr_10671]	ReceptionComSpecProps.comHandlerTaskMappingEnabled vs. implicit data read access
[constr_10672]	Applicability of transportErrorCountEnabled, valueErrorCountEnabled, returnNoNewDataEnabled
[constr_10673]	Existence of ReceptionComSpecProps.comHandlerTaskMappingEnabled vs. ReceptionComSpecProps.timeout
[constr_10674]	Existence of attribute ReceptionComSpecProps.outOfRangeBitfieldErrorsEnabled
[constr_10675]	Existence of attribute ReceptionComSpecProps.invalidValueBitfieldErrorsEnabled
[constr_10676]	Allowed attribute of attribute DiagnosticExtendedDataRecordElement.swDataDefProps
[constr_10677]	Allowed values for attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize
[constr_10678]	Existence of DiagnosticEdrSenderPortMapping.dataPrototype
[constr_10679]	Existence of DiagnosticEdrSenderPortMapping.recordElement
[constr_10680]	Existence of DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem vs. bswServiceDependency
[constr_10681]	Existence of DiagnosticEdrServerPortMapping.recordElement
[constr_10687]	Existence of reference ISignalPortToDiagnosticEventMapping.iSignalTriggering
[constr_10688]	Existence of reference ISignalPortToDiagnosticEventMapping.iSignalPort
[constr_10689]	Restriction for attribute communicationDirection of an ISignalPort that is referenced by ISignalPortToDiagnosticEventMapping
[constr_13000]	Existence of reference in the role EcucContainerDef.destinationUri for an EcucContainerDef aggregated in the role EcucDestinationUriPolicy.container
[constr_2641]	Mixed content elements cannot be split into different partial models
[constr_3796]	Applicable attributes of IEEE1722TpIidcConnection for category IIDC
[constr_3797]	Applicable attributes of IEEE1722TpIidcConnection for category IEC61883
[constr_3799]	DDS Fire & Forget Method definition
[constr_3800]	Existence of DdsCpServiceInstance.discoveryType
[constr_3801]	Existence of DdsCpServiceInstance.serviceInterfaceId
[constr_3802]	Existence of DdsCpServiceInstance.majorVersion

▽

△

Number	Heading
[constr_3803]	Existence of DdsCpServiceInstance.serviceMinorVersion
[constr_3804]	Existence of DdsCpServiceInstance.transportProtocol
[constr_3805]	Existence of DdsServiceInstanceEventCp.eventTriggering
[constr_3806]	Existence of DdsServiceInstanceOperationCp.operationRequestTriggering
[constr_3807]	Existence of DdsServiceInstanceFieldCp reference
[constr_3808]	Existence of DdsCpProvidedServiceInstance.providedServiceInstanceId
[constr_3809]	Existence of DdsCpConsumedServiceInstance.requiredServiceInstanceId
[constr_3810]	Value range of DdsCpProvidedServiceInstance.providedServiceInstanceId
[constr_3811]	DdsCpProvidedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_3812]	DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast
[constr_3813]	DdsCpProvidedServiceInstance unicast address references target
[constr_3814]	Multiple SocketAddress entries with the same IP Address and Port in the context of a given EcuInstance
[constr_3815]	DDS Service communication is restricted to one VLAN
[constr_3816]	Value range of DdsCpConsumedServiceInstance.requiredServiceInstanceId
[constr_3817]	DdsCpConsumedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_3818]	DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast
[constr_3819]	DdsCpConsumedServiceInstance unicast address references target
[constr_3820]	DDS Service communication is restricted to one VLAN
[constr_3821]	DDS transformer configuration
[constr_3822]	No further transfomers in scope of the DDS transformer
[constr_3824]	No explicitWakeupChannel reference for channelSynchronousWakeup = TRUE
[constr_9349]	An ISignal shall not be used for SOME/IP and DDS transport at the same time
[constr_9354]	Definition of unconditional IPdu Timing
[constr_9355]	Category of HwElement for EcuPartitionToCoreMapping
[constr_9357]	Existence of attributes of McDataInstance depending on the category
[constr_9358]	ISignal that references a DataTransformation shall have the dataTypePolicy set to transformingISignal
[constr_9359]	Different EcuPartitionToCoreMapping.coreId attribute values shall not be assigned to the same processingUnit
[constr_9360]	DataMapping not available for elements of PortInterfaces with isService = true
[constr_9361]	FlatInstanceDescriptor.dataPrototype reference restriction

▽

△

Number	Heading
[constr_9362]	Mutually exclusive existence of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation vs. FlatInstanceDescriptor.ecuExtractReference
[constr_9363]	FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation
[constr_9366]	GlobalTimeSlave.followUpTimeoutValue not applicable to GlobalTimeFrSlave
[constr_9367]	Definition of several local ApplicationEndpoints with identical properties (IP, Port, Protocol) is forbidden
[constr_9368]	Applicable attributes for an Ethernet switch
[constr_9369]	Applicable attributes for an Ethernet node on the classic platform
[constr_9371]	Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a RPortPrototype mapped to the SystemSignal
[constr_9372]	Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a PPortPrototype mapped to the SystemSignal
[constr_9373]	Compatibility of DataConstr between SystemSignal and DataPrototype in a PPortPrototype
[constr_9374]	Compatibility of DataConstr between SystemSignal and DataPrototype in a RPortPrototype
[constr_9375]	Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a PPortPrototype mapped to the SystemSignal
[constr_9376]	Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a RPortPrototype mapped to the SystemSignal
[constr_9377]	J1939Node.pduTriggering reference restriction
[constr_9378]	J1939Nodes on the same EcuInstance shall not share PduTriggerings
[constr_9379]	Consistent mapping of software-component to J1939Node
[constr_9380]	Consistent mapping of J1939ControllerApplication to EcuInstance
[constr_9381]	J1939Node shall be referenced by a single J1939ControllerApplicationToJ1939NodeMapping
[constr_9382]	AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy shall only target AppTaskProxy
[constr_9383]	AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy shall only target EcuTaskProxy
[constr_9384]	Relation between both CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime and CyclicHandlingComDataToOsTaskProxyMapping.offset and OsTaskProxy.period
[constr_9385]	Existence of CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy
[constr_9386]	Existence of CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype

▽

△

Number	Heading
[constr_9387]	One CyclicHandlingComDataToOsTaskProxyMapping for every VariableDataPrototype with ReceptionComSpecProps.comHandlerTaskMappingEnabled = TRUE
[constr_9388]	CanNmNode.txNmPdu reference restriction
[constr_9389]	UdpNmNode.txNmPdu reference restriction
[constr_9390]	J1939NmNode.txNmPdu reference restriction
[constr_9391]	FlexrayNmNode.txNmPdu reference restriction
[constr_9395]	Each call ISignal shall have exactly one corresponding return ISignal that is associated with the same ClientServerOperation and PhysicalChannel
[constr_9396]	SomeipTpConnection.tpConcurrentProcessingSdu reference restriction
[constr_9397]	Exclusive usage of tpSdu or tpConcurrentProcessingSdu

Table B.4: Added Constraints in R25-11

B.2.2 Changed Constraints in R25-11

Number	Heading
[constr_10044]	Existence of attribute DiagnosticCommonProps.occurrenceCounterProcessing
[constr_10072]	Allowed multiplicities of SenderComSpec attributes for communication between NvBlockSwComponentType and ApplicationSwComponentType
[constr_10089]	Existence of attribute DiagnosticCommonProps.eventCombinationReportingBehavior
[constr_1022]	Limits shall be defined for each direction of CompuMethod
[constr_10368]	Restriction regarding the reference DiagnosticDataIdentifierSet.diagnosticDataIdentifier
[constr_1050]	Extended compatibility of ImplementationDataTypes
[constr_1051]	Compatibility of SwDataDefProps
[constr_1311]	Appearance of safety-related possible values of SwAddrMethod.option
[constr_1375]	Existence of attributes of CompuMethod and related meta-classes depending on the value of the category
[constr_1381]	Appearance of core-related possible values of SwAddrMethod.option
[constr_1402]	Applicability of core-related possible values of SwAddrMethod.option related to SwAddrMethod.sectionInitializationPolicy
[constr_1435]	Debouncing in the presence of a DiagnosticEventPortMapping
[constr_1669]	Existence of PduTriggering.secOcCryptoServiceMapping
[constr_1756]	Existence of attributes DiagnosticExtendedDataRecord.trigger and update
[constr_1760]	Existence of DiagnosticExtendedDataRecord.element
[constr_1768]	Existence of attribute DiagnosticEvent.associatedEventIdentification

▽

△

Number	Heading
[constr_1847]	Existence of reference `DiagnosticEdrDataProviderMapping.extendedDataRecordElement`
[constr_2502]	Merged model shall be compliant to the meta-model
[constr_2515]	Categories of packages shall not conflict
[constr_2549]	Units of input axis shall be consistent
[constr_2550]	Units of value axis shall be consistent
[constr_2630]	M1 elements with same identity but different type are not allowed
[constr_3021]	Mapping of `SensorActuatorSwComponent`s to SensorActuator `HwElement`s
[constr_3060]	Allowed Attributes for `networkRepresentationProps` and `physicalProps`
[constr_3082]	Value of category in `GeneralPurposeIPdu`
[constr_3096]	Allowed values for `diagnosticMessageType`
[constr_3102]	Restriction on usage of `J1939NodeName` attributes
[constr_3120]	Applicable attributes when `destinationUriNestingContract` is set to `targetContainer`
[constr_3215]	`TransformationTechnology.version` and `TransformationTechnology.protocol` settings for request and response of a client/server communication on the same `TransformationTechnology`
[constr_3249]	Category of `HwElement` for `SwcToEcuMapping`
[constr_3263]	Restriction of usage of `SwcToEcuMapping` in a `System`
[constr_3266]	`TransformationTechnology.hasInternalState` setting for a SOME/IP Transformer
[constr_3739]	Value of `ISignal.dataTypePolicy` for all `ISignal`s associated with a `DdsCpServiceInstance`
[constr_3740]	Existence of `DdsServiceInstanceEventCp.ddsEventTopic`
[constr_4526]	Specifying `maxCycleRepetitions` and `maxSlotsPerCycle` in a Repetitive Execution Order Constraint
[constr_4529]	Number of nested elements referenced by the *root* `EOCExecutableEntityRefGroup`
[constr_4531]	Number of nested elements referenced by `EOCExecutableEntityRefGroup` representing a cycle
[constr_4540]	`maxCycleRepetitions` and `maxSlotsPerCycle` shall not be zero
[constr_5273]	One `ISignalTriggering` pair allowed per `EthernetPhysicalChannel` for a `ClientServerOperation`
[constr_5378]	`PduTriggering` shall only be referenced once from a `SomeipTpConnection` in the role `tpSdu` or `tpConcurrentProcessingSdu`
[constr_5521]	`multiplicityConfigClass` attribute of `symbolicNameValue` parameters
[constr_5522]	`postBuildVariantMultiplicity` attribute of `symbolicNameValue` parameters
[constr_6900]	Dual existence of `TDEventVfb.portPrototype` and `TDEventVfb.portPrototypeBlueprint`
[constr_6914]	Restriction of the `portPrototype` context of an `AgeConstraint`

Table B.5: Changed Constraints in R25-11

B.2.3 Deleted Constraints in R25-11

Number	Heading
[constr_1000]	End-to-end protection is limited to sender/receiver communication
[constr_1001]	Value of `dataId` shall be unique
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_1025]	Avoid division by zero in rational formula
[constr_10371]	`DiagnosticDataElement`s owned by a `DiagnosticExtendedDataRecord`
[constr_1040]	Conversion of `SenderReceiverInterface`s
[constr_1041]	Conversion of `ClientServerInterface`s
[constr_10548]	Uniqueness of `ReceiverComSpec.dataElement`
[constr_10549]	Uniqueness of `SenderComSpec.dataElement`
[constr_10550]	Uniqueness of `ClientComSpec.operation`
[constr_10551]	Uniqueness of `ServerComSpec.operation`
[constr_10552]	Uniqueness of `ModeSwitchSenderComSpec.modeGroup`
[constr_10553]	Uniqueness of `ModeSwitchReceiverComSpec.modeGroup`
[constr_10554]	Uniqueness of `ParameterProvideComSpec.parameter`
[constr_10555]	Uniqueness of `ParameterRequireComSpec.parameter`
[constr_10556]	Uniqueness of `NvRequireComSpec.variable`
[constr_10557]	Uniqueness of `NvProvideComSpec.variable`
[constr_1111]	Constraints of `dataId` in `PROFILE_01`
[constr_1112]	Constraints of `dataIdMode` in `PROFILE_01`
[constr_1113]	Existence of attributes of meta-class `EndToEndDescription` in `PROFILE_01`
[constr_1114]	Constraints of `crcOffset` in `PROFILE_01`
[constr_1115]	Constraints of `counterOffset` in `PROFILE_01`
[constr_1116]	Constraints of `dataLength` in `PROFILE_01`
[constr_1117]	Constraints of `maxDeltaCounterInit` in `PROFILE_01`
[constr_1118]	Existence of attributes of meta-class `EndToEndDescription` in `PROFILE_02`
[constr_1119]	Constraints of `dataLength` in `PROFILE_02`
[constr_1120]	Constraints of `dataId` in `PROFILE_02`
[constr_1121]	Constraints of `maxDeltaCounterInit` in `PROFILE_02`
[constr_1170]	Existence of attribute `EndToEndDescription.maxDeltaCounterInit` for `PROFILE_01`
[constr_1171]	Existence of attribute `EndToEndDescription.maxDeltaCounterInit` for `PROFILE_02`
[constr_1183]	`EndToEndProtectionVariablePrototype`s aggregated by `EndToEndProtection`
[constr_1211]	Constraints of `maxNoNewOrRepeatedData` in `PROFILE_01`
[constr_1212]	Constraints of `syncCounterInit` in `PROFILE_01`
[constr_1213]	Constraints of `maxNoNewOrRepeatedData` in `PROFILE_02`

▽

△

Number	Heading
[constr_1214]	Constraints of `syncCounterInit` in `PROFILE_02`
[constr_1215]	Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_01`
[constr_1216]	Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_01`
[constr_1217]	Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_02`
[constr_1218]	Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_02`
[constr_1261]	Applicability for `EndToEndDescription.dataIdNibbleOffset`
[constr_1463]	Applicable values for `J1939Cluster.networkId`
[constr_1612]	Reference from `DiagnosticRoutineControl` to `DiagnosticAccessPermission` has no meaning
[constr_1761]	Existence of attribute `DiagnosticConnectedIndicator.healingCycle`
[constr_1782]	Usage of internal data elements only for extended data records
[constr_1901]	Existence of attribute `EndToEndDescription.category`
[constr_1902]	Existence of attribute `EndToEndProtection.endToEndProfile`
[constr_2510]	Only one default `ReferenceBase`
[constr_2548]	Data constraint of value axis shall match
[constr_2573]	ICS shall not reference examples
[constr_2608]	Custom extensions shall be part of the `Documentation` that is referenced by the `Baseline`
[constr_2609]	Single revision per AUTOSAR standard
[constr_2610]	No alternativeName if matching via `shortName`
[constr_2611]	Referenced AUTOSAR Specification Elements shall be part of the AUTOSAR Specification Baseline
[constr_2612]	`shortName` of `ConcreteClassTailoring` shall match the name of an AUTOSAR specified concrete meta-class
[constr_2613]	`shortName` of `AbstractClassTailoring` shall match the name of an AUTOSAR specified abstract meta-class
[constr_2614]	`PrimitiveAttributeCondition.attribute` shall reference invariant owned `PrimitiveAttributeTailoring`, only
[constr_2615]	`AggregationCondition.aggregation` shall reference invariant owned `AggregationTailoring`, only
[constr_2616]	`ReferenceCondition.reference` shall reference invariant owned `ReferenceTailoring`, only
[constr_2617]	`ClassTailoring.variationRestriction` only applicable for «atpVariation» classes
[constr_2618]	ShortName of AttributeTailoring shall match owned or inherited attributes
[constr_2619]	No `AttributeTailoring` for Derived or Abstract Attributes
[constr_2620]	`shortName` of `PrimitiveAttributeTailoring` shall be a primitive attribute in the referenced Baseline

▽

△

Number	Heading
[constr_2621]	The shortName of AggregationTailoring shall match the name of an AUTOSAR specified aggregation of the meta-class
[constr_2622]	The shortName of ReferenceTailoring shall match the name of an AUTOSAR specified reference of the meta-class
[constr_2623]	Referenced SdgClass shall be part of a SdgDef that is referenced by the Baseline
[constr_2624]	AttributeTailoring.variationRestriction only applicable for «atp Variation» attributes
[constr_2632]	No postbuild variation for attribute values
[constr_3119]	Necessary content of EcucDestinationUriDefs that are referenced by an EcucContainerDef
[constr_3239]	Consistent mapping of software-component to J1939NmNode
[constr_3240]	Consistent mapping of J1939ControllerApplication to EcuInstance
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3299]	SocketAddress.pathMtuDiscoveryEnabled setting dependency
[constr_3435]	Applicability of CouplingPort.macMulticastAddress
[constr_3509]	Applicability of scope attribute
[constr_3738]	ISignal that has dataTypePolicy set to ddsSignal or to ddsService shall not reference a DataTransformation
[constr_3779]	Number of ISignal.receptionDefaultValue elements
[constr_3780]	ISignal.receptionDefaultValue definition in case that the SOME/IP Serializer receives less data than expected
[constr_4055]	ICS may not contain blueprints
[constr_5029]	J1939NmCluster is not allowed to reference a TtcanCluster
[constr_5459]	Existence of dataLength
[constr_5460]	Existence of policyAction
[constr_5461]	Existence of timeInterval
[constr_6848]	Existence of VfbTiming.component
[constr_6878]	Existence of TDEventTTCanCycleStart.ttCanCluster
[constr_9112]	Existence of GenericTp.tpTechnology
[constr_9115]	Existence of RtpTp.ssrc
[constr_9116]	Existence of RtpTp.tcpUdpConfig
[constr_9119]	Existence of Ieee1722Tp.streamIdentifier
[constr_9120]	Existence of HttpTp.protocolVersion
[constr_9121]	Existence of HttpTp.tcpTpConfig
[constr_9207]	Existence of EndToEndProtectionISignalIPdu.iSignalIPdu
[constr_9208]	Existence of EndToEndProtectionISignalIPdu.iSignalGroup
[constr_9209]	Existence of EndToEndProtectionISignalIPdu.dataOffset
[constr_9267]	Existence of J1939TpConnection.broadcast
[constr_9272]	Existence of TlsPskIdentity.preSharedKey

▽

△

Number	Heading
[constr_9309]	Existence of GlobalTimeCanMaster.syncConfirmationTimeout

Table B.6: Deleted Constraints in R25-11

B.3 Traceable item history of this document according to AUTOSAR Release R24-11

B.3.1 Added Constraints in R24-11

Number	Heading
[constr_10520]	Multiplicity of AssemblySwConnector.provider
[constr_10521]	Multiplicity of AssemblySwConnector.requester
[constr_10522]	OBD trouble code shall only be placed in primary fault memory
[constr_10523]	Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds
[constr_10524]	Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeObd
[constr_10525]	Existence of attribute ApplicationValueSpecification.category
[constr_10527]	Existence of RoleBasedDataAssignment.usedDataElement.autosarVariable for RoleBasedDataAssignment.role = ramBlock
[constr_10529]	Existence of AsynchronousServerCallResultPoint for AsynchronousServerCallPoint where attribute timeout is defined
[constr_10532]	Restriction for SenderComSpec. transmissionProps.onChangeDataPrototype
[constr_10533]	Existence of TransmissionComSpecProps.onChangeDataPrototype.data-PrototypeInSenderReceiverInterface.rootDataPrototypeInSr
[constr_10534]	Existence of TransmissionComSpecProps. onChangeDataPrototype.rootDataPrototype
[constr_10538]	Existence of attribute ReceiverComSpec.dataElement
[constr_10539]	Existence of attribute SenderComSpec.dataElement
[constr_10542]	RunnableEntity is referenced by an OperationInvokedEvent
[constr_10543]	Uniqueness of reference PortAPIOption.port
[constr_10544]	Ownership of reference PortAPIOption.port
[constr_10545]	Existence of DiagnosticParameterIdentifier.dataElement.
[constr_10548]	Uniqueness of ReceiverComSpec.dataElement
[constr_10549]	Uniqueness of SenderComSpec.dataElement
[constr_10550]	Uniqueness of ClientComSpec.operation
[constr_10551]	Uniqueness of ServerComSpec.operation

▽

△

Number	Heading
[constr_10552]	Uniqueness of ModeSwitchSenderComSpec.modeGroup
[constr_10553]	Uniqueness of ModeSwitchReceiverComSpec.modeGroup
[constr_10554]	Uniqueness of ParameterProvideComSpec.parameter
[constr_10555]	Uniqueness of ParameterRequireComSpec.parameter
[constr_10556]	Uniqueness of NvRequireComSpec.variable
[constr_10557]	Uniqueness of NvProvideComSpec.variable
[constr_10558]	SwBaseType associated with corresponding ApplicationRecordElement and ImplementationDataTypeElement
[constr_10559]	Uniqueness of DataPrototypeMapping.firstDataPrototype and secondDataPrototype
[constr_10560]	Uniqueness of ClientServerOperationMapping.firstOperation and secondOperation
[constr_10561]	Uniqueness of ClientServerApplicationErrorMapping.firstApplicationError and secondApplicationError
[constr_10562]	Uniqueness of ModeDeclarationGroupPrototypeMapping.firstModeGroup and secondModeGroup
[constr_10563]	Uniqueness of ModeDeclarationMapping.firstMode and secondMode
[constr_10564]	Uniqueness of TriggerMapping.firstTrigger and secondTrigger
[constr_10565]	Uniqueness of SubElementMapping.firstElement and secondElement
[constr_10573]	Existence of attribute DiagnosticServiceTable.diagnosticServiceInstance
[constr_10575]	No multiple instantiation of NvBlockSwComponentType
[constr_10577]	Existence of DiagnosticResponseOnEventClass.storeEventEnabled
[constr_10606]	Existence of ConstantSpecificationMapping or CalibrationParameterValue for ApplicationValueSpecification or ApplicationRuleBasedValueSpecification of category CURVE, MAP, CUBOID, CUBE_4, and CUBE_5
[constr_10607]	Number of ConstantSpecificationMappings that are allowed to reference a ApplicationValueSpecification or ApplicationRuleBasedValueSpecification in the context of an InternalBehavior
[constr_10608]	Number of ConstantSpecificationMappings that are allowed to reference a ApplicationValueSpecification or ApplicationRuleBasedValueSpecification in the context of a ParameterSwComponentType
[constr_10610]	Compatibility of PhysicalDimensions in the context is the creation of an ApplicationValueSpecification
[constr_3763]	Allowed value for maxDeltaCounter in the context of a profileName
[constr_3764]	Applicability of CouplingPort.macAddressVlanAssignment
[constr_3765]	Applicability of MacAddressVlanMembership.vlan
[constr_3766]	Valid MacAddressVlanMembership.vlan target EthernetPhysicalChannel
[constr_3767]	NmNode.nmVariant setting to slavePassive

▽

△

Number	Heading
[constr_3768]	NmNode.nmVariant setting to slaveActive
[constr_3769]	NmNode.nmVariant setting to full
[constr_3770]	NmNode.nmVariant setting to passive
[constr_3771]	Range of NmCluster.nmLightTimeout
[constr_3779]	Number of ISignal.receptionDefaultValue elements
[constr_3780]	ISignal.receptionDefaultValue definition in case that the SOME/IP Serializer receives less data than expected
[constr_3781]	Each PNC assigned to multiple PhysicalChannels shall have a top level PNC-Coordinator
[constr_3782]	Consistent framePreemptionSupport setting in the scope of one CouplingPortConnection
[constr_3783]	Definition of CouplingPortFifo.trafficClassPreemptionSupport only in context of an Ethernet switch
[constr_3784]	Applicable CouplingPortFifo as predecessor for portScheduler = enhancedTrafficShaper
[constr_3785]	Exclusive definition of etsAvailableBandwidthInPercent or etsAvailableBandwidthInWeightValue
[constr_3786]	Consistent usage of either etsAvailableBandwidthInPercent or etsAvailableBandwidthInWeightValue for portScheduler = enhancedTrafficShaper
[constr_3787]	Existence of CouplingPortTrafficClassAssignment.trafficClass
[constr_3788]	Existence of CouplingPortFifo.assignedTrafficClass
[constr_3789]	Allowed values for CouplingPortFifo.assignedTrafficClass
[constr_3790]	Existence of CouplingPortDetails.defaultTrafficClass
[constr_3791]	Allowed values for CouplingPortDetails.defaultTrafficClass
[constr_3792]	FrameMapping between identical bus systems
[constr_3793]	Usage of KeepLocalPduBuffer
[constr_3794]	Usage of PduBufferAlignment
[constr_6918]	Referenced TimingDescriptions in TDCpSoftwareClusterMapping and TDCpSoftwareClusterResourceMapping
[constr_6919]	Referenced CpSoftwareCluster of TDCpSoftwareClusterMapping
[constr_6920]	Existence of LatencyTimingConstraint.minimum used in an LET interval
[constr_6921]	Disallow TimingDescriptionEventChain segmental circular-referencing
[constr_9316]	Multi instantiated BSW Modules not mappable
[constr_9317]	StateDependentFirewall.firewallStateModeDeclaration reference restriction
[constr_9318]	Reception of CanFrameTriggerings with the same identifier by an EcuInstance
[constr_9319]	Value of BusMirrorChannelMappingCan.mirroringProtocol
[constr_9320]	Value of BusMirrorChannelMappingFlexray.mirroringProtocol
[constr_9321]	Same time base for all BusMirrorChannelMappings of one EcuInstance

▽

△

Number	Heading
[constr_9326]	Exclusive existence of ISignalTriggering.iSignal and ISignalTriggering.iSignalGroup
[constr_9330]	Derivation of network representation in case that several DataMappings are defined that map the same SystemSignal to different VariableDataPrototypes
[constr_9331]	E2E protection of a ClientServerOperation
[constr_9332]	Existence of J1939TpConnection.tpProtocolType
[constr_9333]	FibexElements in ECU_EXTRACT
[constr_9343]	Allowed J1939ProtectedIPdu.payload reference target
[constr_9346]	Existence of EthernetVlanTranslationTable.translatedVlanId
[constr_9347]	Range of EthernetVlanTranslationTable.ingressVlanId and EthernetVlanTranslationTable.translatedVlanId
[constr_9348]	EthernetVlanTranslationTable.translatedVlanId and vlanMembership

Table B.7: Added Constraints in R24-11

B.3.2 Changed Constraints in R24-11

Number	Heading
[constr_1000]	End-to-end protection is limited to sender/receiver communication
[constr_1001]	Value of dataId shall be unique
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_10024]	Existence of reference in the role DiagnosticSecurityEventReportingModeMapping.dataElement
[constr_1007]	Allowed attributes of SwDataDefProps for ApplicationDataTypes
[constr_10088]	Relation between event and DTC without event combination
[constr_10099]	Allowed values of the attribute SwDataDefProps.swImplPolicy vs. DataPrototypes and their roles
[constr_1012]	Value of category is FIXED_LENGTH
[constr_1053]	Compatibility of PhysicalDimensions in the context of the creation of a SwConnector
[constr_1070]	Compatibility of PortPrototypes of different DataInterfaces in the context of DelegationSwConnectors
[constr_1087]	AssemblySwConnector inside CompositionSwComponentType
[constr_1088]	DelegationSwConnector inside CompositionSwComponentType
[constr_1111]	Constraints of dataId in PROFILE_01
[constr_1112]	Constraints of dataIdMode in PROFILE_01
[constr_1113]	Existence of attributes of meta-class EndToEndDescription in PROFILE_01
[constr_1114]	Constraints of crcOffset in PROFILE_01

▽

△

Number	Heading
[constr_1115]	Constraints of `counterOffset` in `PROFILE_01`
[constr_1116]	Constraints of `dataLength` in `PROFILE_01`
[constr_1117]	Constraints of `maxDeltaCounterInit` in `PROFILE_01`
[constr_1118]	Existence of attributes of meta-class `EndToEndDescription` in `PROFILE_02`
[constr_1119]	Constraints of `dataLength` in `PROFILE_02`
[constr_1120]	Constraints of `dataId` in `PROFILE_02`
[constr_1121]	Constraints of `maxDeltaCounterInit` in `PROFILE_02`
[constr_1170]	Existence of attribute `EndToEndDescription.maxDeltaCounterInit` for `PROFILE_01`
[constr_1171]	Existence of attribute `EndToEndDescription.maxDeltaCounterInit` for `PROFILE_02`
[constr_1183]	`EndToEndProtectionVariablePrototype`s aggregated by `EndToEndProtection`
[constr_1211]	Constraints of `maxNoNewOrRepeatedData` in `PROFILE_01`
[constr_1212]	Constraints of `syncCounterInit` in `PROFILE_01`
[constr_1213]	Constraints of `maxNoNewOrRepeatedData` in `PROFILE_02`
[constr_1214]	Constraints of `syncCounterInit` in `PROFILE_02`
[constr_1215]	Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_01`
[constr_1216]	Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_01`
[constr_1217]	Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_02`
[constr_1218]	Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_02`
[constr_1261]	Applicability for `EndToEndDescription.dataIdNibbleOffset`
[constr_1295]	`PortInterface`s and `category` `DATA_REFERENCE`
[constr_1349]	Value of `udsDtcValue` shall be unique
[constr_1363]	Existence of attributes of `DiagnosticValueNeeds`
[constr_1519]	Existence of attributes vs. `category` of `ApplicationValueSpecification`
[constr_1768]	Existence of attribute `DiagnosticEvent.associatedEventIdentification`
[constr_1829]	Existence of reference `DiagnosticConnectedIndicator.indicator`
[constr_1901]	Existence of attribute `EndToEndDescription.category`
[constr_1902]	Existence of attribute `EndToEndProtection.endToEndProfile`
[constr_1986]	Existence of the reference `DiagnosticRoutineNeeds.diagRoutineType`
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3163]	`EndToEndTransformationISignalProps.minDataLength` and `EndToEndTransformationISignalProps.maxDataLength` in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76

▽

△

Number	Heading
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76
[constr_3169]	EndToEndTransformationDescription.offset value in PROFILE_02, PROFILE_22 and PROFILE_76
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76 or dataIdMode different from lower12Bit
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3267]	PduTriggerings in Service Discovery StaticSocketConnections
[constr_3268]	Service Discovery StaticSocketConnection aggregation by a SocketAddress
[constr_3269]	Service Discovery StaticSocketConnection remoteAddress reference to a TpPort
[constr_3270]	Service Discovery SocketConnection remoteAddress reference to an IP Address
[constr_3272]	SoConIPduIdentifier.headerId setting for SD StaticSocketConnections
[constr_3273]	Service Discovery multicast StaticSocketConnection's aggregation by an ApplicationEndpoint

▽

△

Number	Heading
[constr_3274]	Service Discovery unicast StaticSocketConnection's aggregation by an ApplicationEndpoint
[constr_3519]	Value of category of GlobalTimeDomain
[constr_3600]	Setting of EthernetCommunicationController.slaveActAsPassive-CommunicationSlave
[constr_3697]	Latest existence time of CanControllerXlConfiguration and CanControllerXlConfigurationRequirements
[constr_3714]	Multiple top level PNC-coordinators shall be allowed
[constr_3716]	SecuredIPdu.dynamicRuntimeLengthHandling for dynamic length Pdus
[constr_4014]	Call type and execution context
[constr_4016]	BswCalledEntity constraints
[constr_4018]	BswInterruptEntity constraints
[constr_4071]	Synchronized runnables and schedulable entities shall be consistent
[constr_4089]	Association callbackHeader is only applicable for BSW modules
[constr_4090]	The callbackHeader reference has to be consistent with behavior reference
[constr_4102]	Semantics of McGroupDataRefSet.mcDataInstance
[constr_4500]	Restricted usage of Occurence Expression functions
[constr_4502]	Use references only as function operands
[constr_4503]	Restricted usage of AutosarOperationArgumentInstance for Content Filter
[constr_4504]	Restriction of the scope of an AgeConstraint
[constr_4505]	Specifying minimum and maximum number of occurrences
[constr_4506]	Specifying minimum inter-arrival time and pattern length
[constr_4507]	Specifying pattern length, pattern jitter and patter period
[constr_4508]	Existence of TDEventVfbPort.portPrototypeBlueprint
[constr_4510]	Specifying references to RunnableEntity and VariableAccess
[constr_4511]	Validity of referencing RunnableEntity
[constr_4512]	Validity of referencing VariableAccess
[constr_4513]	SynchronizationTimingConstraint shall reference at least two events
[constr_4514]	SynchronizationTimingConstraint shall reference at least two event chains
[constr_4515]	Orthogonality of stimulus and response in a TimingDescriptionEventChain
[constr_4516]	Completeness of a composed TimingDescriptionEventChain
[constr_4518]	Specifying end-points of a composed TimingDescriptionEventChain
[constr_4519]	Specifying patternLength
[constr_4520]	Specifying attribute synchronizationConstraintType
[constr_4521]	Specifying attribute synchronizationConstraintType
[constr_4522]	SynchronizationTimingConstraint shall either reference events or event chains
[constr_4523]	Restriction of maxCycleRepetitions and maxSlotsPerCycle to Repetitive Execution Order Constraint

▽

△

Number	Heading
[constr_4525]	Precedence of successor relationships `successor` and `directSuccessor`
[constr_4526]	Specifying `maxCycles` and `maxSlots` in a Repetitive Execution Order Constraint
[constr_4527]	Referencing `TimingDescriptionEvent` in a Repetitive Execution Order Constraint
[constr_4528]	The *root* `EOCExecutableEntityRefGroup` shall reference only `EOCExecutableEntityRefGroup`s
[constr_4529]	Number of nested elements referenced by the *root* `EOCExecutableEntityRefGroup`
[constr_4530]	An `EOCExecutableEntityRefGroup` representing a cycle shall reference only `EOCExecutableEntityRef`s respectively `EOCEventRef`s
[constr_4531]	Number of nested elements referenced by `EOCExecutableEntityRefGroup` representing a cycle
[constr_4532]	Successor relationship is not self-referencing
[constr_4533]	Maximum number of successor relationships
[constr_4534]	Maximum number of `directSuccessor` relationships
[constr_4536]	Compatible recurrence of any `ExecutableEntity`
[constr_4537]	References among elements in an `ExecutionOrderConstraint`
[constr_4538]	Hierarchical Execution Order Constraint: `EOCExecutableEntityRef`, `EOCEventRef`, and `EOCExecutableEntityRefGroup` shall be target or source of a successor relationship
[constr_4539]	The successor relationships `successor` and `directSuccessor` shall not be used
[constr_4540]	`maxCycles` and `maxSlots` shall not be zero
[constr_4541]	Existence of `EOCExecutableEntityRef.executable` in an Ordinary Execution Order Constraint
[constr_4542]	Existence of `EOCExecutableEntityRef.executable` in a Hierarchical Execution Order Constraint
[constr_4543]	Maximum value of `minimumInterArrivalTime`
[constr_4544]	Specifying `patternLength`, `patternJitter` and `patternPeriod`
[constr_4545]	Referring either `ExecutableEntity`s or `AbstractEvent`s
[constr_4546]	Setting the attribute `isEvent`
[constr_4547]	Restriction of `ExecutionOrderConstraint.permitMultipleReferencesToEE`
[constr_4548]	Existence of `EOCEventRef.event` in an Ordinary Execution Order Constraint
[constr_4549]	Existence of `EOCEventRef.event` in a Hierarchical Execution Order Constraint
[constr_4551]	Use only Numericals in `TDEventOccurrenceExpression`
[constr_4552]	Restricted usage of `AutosarVariableInstance` for Content Filter
[constr_4554]	Restriction of the referenced `TimingDescriptionEventChain` for a `letInterval`
[constr_4559]	Restriction of `TimingDescriptionEvent.category`
[constr_4561]	Usage of the category value `DISPATCH_ENTRY_POINT` in `TimingDescriptionEvent`

▽

△

Number	Heading
[constr_4562]	Usage of the category value DISPATCH_EXIT_POINT in TimingDescriptionEvent
[constr_4565]	Consistency of TDCpSoftwareClusterMapping.timingDescription and TDCpSoftwareClusterResourceMapping.timingDescription
[constr_5049]	Ethernet switch packet to traffic class assignment restriction
[constr_5091]	Relevance of tcpRole attribute
[constr_5105]	Mapping of Pdu with dynamic length in a FlexrayFrame
[constr_5168]	pncGatewayType passive and connected ECUs
[constr_5221]	Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, and PROFILE_76
[constr_5273]	One ISignalTriggering pair allowed per EthernetPhysicalChannel for a ClientServerOperation
[constr_5380]	Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of an EcuInstance
[constr_6816]	Restricted usage of TimingDescriptionEventChain.isPipeliningPermitted in TimingDescriptionEventChain
[constr_6817]	Restricted usage of TimingDescriptionEvent.clockReference
[constr_6818]	Existence of EventTriggeringConstraint.event
[constr_6819]	Existence of PeriodicEventTriggering.jitter
[constr_6820]	Existence of PeriodicEventTriggering.minimumInterArrivalTime
[constr_6821]	Existence of PeriodicEventTriggering.period
[constr_6822]	Existence of SporadicEventTriggering.maximumInterArrivalTime
[constr_6823]	Existence of SporadicEventTriggering.minimumInterArrivalTime
[constr_6824]	Existence of ConcretePatternEventTriggering.patternLength
[constr_6825]	Existence of BurstPatternEventTriggering.maxNumberOfOccurrences
[constr_6826]	Existence of BurstPatternEventTriggering.minimumInterArrivalTime
[constr_6827]	Existence of BurstPatternEventTriggering.patternLength
[constr_6828]	Existence of ArbitraryEventTriggering.minimumDistance
[constr_6829]	Existence of ArbitraryEventTriggering.maximumDistance
[constr_6830]	Existence of ConfidenceInterval.lowerBound
[constr_6831]	Existence of ConfidenceInterval.propability
[constr_6832]	Existence of ConfidenceInterval.upperBound
[constr_6833]	Existence of ExecutionOrderConstraint.orderedElement
[constr_6834]	Existence of EOCExecutableEntityRefGroup.nestedElement
[constr_6835]	Existence of ExecutionTimeConstraint.executionTimeType
[constr_6836]	Existence of ExecutionTimeConstraint.executable
[constr_6837]	Existence of LatencyTimingConstraint.latencyConstraintType
[constr_6838]	Existence of LatencyTimingConstraint.maximum

▽

△

Number	Heading
[constr_6839]	Existence of LatencyTimingConstraint.minimum
[constr_6841]	Existence of LatencyTimingConstraint.scope
[constr_6842]	Existence of OffsetTimingConstraint.maximum
[constr_6843]	Existence of OffsetTimingConstraint.minimum
[constr_6844]	Existence of OffsetTimingConstraint.source
[constr_6845]	Existence of OffsetTimingConstraint.target
[constr_6846]	Existence of SynchronizationTimingConstraint.synchronizationConstraintType
[constr_6847]	Existence of SynchronizationTimingConstraint.tolerance
[constr_6848]	Existence of VfbTiming.component
[constr_6849]	Existence of SystemTiming.system
[constr_6850]	Existence of BswModuleTiming.behavior
[constr_6851]	Existence of BswCompositionTiming.implementation
[constr_6852]	Existence of EcuTiming.ecuConfiguration
[constr_6853]	Existence of ModeInBswInstanceRef.contextModeDeclarationGroupPrototype
[constr_6854]	Existence of ModeInBswInstanceRef.targetModeDeclaration
[constr_6855]	Existence of ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype
[constr_6856]	Existence of ModeInSwcInstanceRef.contextPort
[constr_6857]	Existence of ModeInSwcInstanceRef.targetModeDeclaration
[constr_6858]	Existence of TDEventBswInternalBehavior.tdEventBswInternalBehaviorType
[constr_6859]	Existence of TDEventBswInternalBehavior.bswModuleEntity
[constr_6860]	Existence of TDEventBswModule.tdEventBswModuleType
[constr_6861]	Existence of TDEventBswModule.bswModuleEntry
[constr_6862]	Existence of TDEventBswModeDeclaration.tdEventBswModeDeclarationType
[constr_6863]	Existence of TDEventBswModeDeclaration.modeDeclaration
[constr_6864]	Existence of TDEventISignal.tdEventType
[constr_6865]	Existence of TDEventISignal.iSignal
[constr_6866]	Existence of TDEventISignal.physicalChannel
[constr_6867]	Existence of TDEventIPdu.tdEventType
[constr_6868]	Existence of TDEventIPdu.iPdu
[constr_6869]	Existence of TDEventIPdu.physicalChannel
[constr_6870]	Existence of TDEventFrame.tdEventType
[constr_6871]	Existence of TDEventFrame.frame
[constr_6872]	Existence of TDEventFrame.physicalChannel
[constr_6873]	Existence of TDEventFrameEthernet.tdEventType
[constr_6874]	Existence of TDHeaderIdRange.maxHeaderId

▽

△

Number	Heading
[constr_6875]	Existence of TDHeaderIdRange.minHeaderId
[constr_6876]	Existence of TDEventCycleStart.cycleRepetition
[constr_6877]	Existence of TDEventFrClusterCycleStart.frCluster
[constr_6878]	Existence of TDEventTTCanCycleStart.ttCanCluster
[constr_6879]	Existence of TDEventOccurrenceExpression.formula
[constr_6880]	Existence of AutosarVariableInstance.variableInstance
[constr_6881]	Existence of AutosarOperationArgumentInstance.operationArgumentInstance
[constr_6882]	Existence of TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType
[constr_6883]	Existence of TDEventSwcInternalBehavior.runnable
[constr_6884]	Existence of TDEventSwcInternalBehaviorReference.referencedTDEventSwc
[constr_6885]	Existence of TDEventVfbPort.isExternal
[constr_6886]	Existence of TDEventVfbReference.referencedTDEventVfb
[constr_6887]	Existence of TDEventVariableDataPrototype.tdEventVariableDataPrototypeType
[constr_6888]	Existence of TDEventVariableDataPrototype.dataElement
[constr_6889]	Existence of TDEventOperation.tdEventOperationType
[constr_6890]	Existence of TDEventOperation.operation
[constr_6891]	Existence of TDEventModeDeclaration.tdEventModeDeclarationType
[constr_6892]	Existence of TDEventModeDeclaration.modeDeclaration
[constr_6893]	Existence of TDEventTrigger.tdEventTriggerType
[constr_6894]	Existence of TDEventTrigger.trigger
[constr_6895]	Existence of TimingDescriptionEventChain.response
[constr_6896]	Existence of TimingDescriptionEventChain.stimulus
[constr_6897]	Existence of TimingDescriptionEventChain.segment
[constr_6898]	Existence of ConcretePatternEventTriggering.offset
[constr_6899]	Existence of ModeInSwcInstanceRef.base
[constr_6900]	Dual existence of TDEventVfb.port and TDEventVfb.portPrototypeBlueprint
[constr_6901]	Existence of TDEventBsw.bswModuleDescription
[constr_6906]	Conformity of stimulus and response in a TimingDescriptionEventChain
[constr_6907]	Restriction of EOCExecutableEntityRefGroup.triggeringEvent
[constr_6908]	Restriction of EOCExecutableEntityRefGroup.letDataExchangeParadigm
[constr_6909]	Singleton ROOT_GROUP in a Hierarchical Execution Order Constraint
[constr_6910]	Referencing **from** a ROOT_GROUP in a Hierarchical Execution Order Constraint
[constr_6911]	Referencing **to** a ROOT_GROUP in a Hierarchical Execution Order Constraint
[constr_6912]	Mandatory specification of LET interval **recurrence**
[constr_6913]	Restriction on RTEEvents used in an LET interval

▽

△

Number	Heading
[constr_6914]	Restriction of the `port` context of an `AgeConstraint`
[constr_6915]	Affinity of `ISignal` in `TDEventISignal`
[constr_6916]	Affinity of `Frame` in `TDEventFrame`
[constr_6917]	Affinity of `IPdu` in `TDEventIPdu`
[constr_9112]	Existence of `GenericTp.tpTechnology`
[constr_9115]	Existence of `RtpTp.ssrc`
[constr_9116]	Existence of `RtpTp.tcpUdpConfig`
[constr_9119]	Existence of `Ieee1722Tp.streamIdentifier`
[constr_9120]	Existence of `HttpTp.protocolVersion`
[constr_9121]	Existence of `HttpTp.tcpTpConfig`
[constr_9207]	Existence of `EndToEndProtectionISignalIPdu.iSignalIPdu`
[constr_9208]	Existence of `EndToEndProtectionISignalIPdu.iSignalGroup`
[constr_9209]	Existence of `EndToEndProtectionISignalIPdu.dataOffset`
[constr_9267]	Existence of `J1939TpConnection.broadcast`

Table B.8: Changed Constraints in R24-11

B.3.3 Deleted Constraints in R24-11

Number	Heading
[constr_10017]	
[constr_1364]	
[constr_1433]	
[constr_1796]	
[constr_1891]	
[constr_2051]	
[constr_2525]	
[constr_2540]	
[constr_3522]	
[constr_4086]	
[constr_4517]	
[constr_4550]	
[constr_4560]	
[constr_4563]	
[constr_4564]	
[constr_4567]	
[constr_4568]	
[constr_5169]	

▽

△

Number	Heading
[constr_5391]	

Table B.9: Deleted Constraints in R24-11