Document Title	Collection of constraints on AUTOSAR M1 models
Document Category	Technical report (TR)
Document UID	635
Document Status	published
Part of AUTOSAR Standard	Foundation
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 5

2 Autosar Model Constraints 6
2.1 FO_TPS_AbstractPlatformSpecification 6
2.2 FO_TPS_FeatureModelExchangeFormat 8
2.3 FO_TPS_LogAndTrace 15
2.4 FO_TPS_SecurityExtractTemplate 18

A Mentioned Class Tables 23

B Change history of AUTOSAR traceable items 52
B.1 Traceable item history of this document according to AUTOSAR Release R26-11 52
B.1.1 Added Constraints in R26-11 52
B.1.2 Changed Constraints in R26-11 52
B.1.3 Deleted Constraints in R26-11 53
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 53
B.2.1 Added Constraints in R25-11 53
B.2.2 Changed Constraints in R25-11 54
B.2.3 Deleted Constraints in R25-11 54
B.3 Traceable item history of this document according to AUTOSAR Release R24-11 54
B.3.1 Added Constraints in R24-11 54
B.3.2 Changed Constraints in R24-11 55
B.3.3 Deleted Constraints in R24-11 55

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 Foundation, 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 FO_TPS_AbstractPlatformSpecification

[constr_6803] Standardized values of `CompositionSwComponentType.category`

Status: DRAFT
Imposition time: `IT_Apsd`

⌈In a `System` with the `category` set to `ABSTRACT_PLATFORM_SYSTEM_DESCRIPTION`, any `CompositionSwComponentType` which is referenced by a `SwComponentPrototype` in the role `type` shall have the `category` set to:

- `XP_COMPONENT_APPLICATION`

⌋

[constr_6806] Standardized values of `ApplicationInterface.category`

Status: DRAFT
Imposition time: `IT_Apsd`

⌈The `category` of a `ApplicationInterface` can be set to either:

- `XP_PORT_SECURITY`
- `XP_PORT_TIMESYNC`
- `XP_PORT_STORAGE`
- `XP_PORT_APPLICATION`
- `XP_PORT_SAFETY`

⌋

[constr_6807] Exclusivity of a `ApplicationInterface` to an Abstract Platform

Status: DRAFT
Imposition time: `IT_Apsd`

⌈A `ApplicationInterface` shall not type a `PortPrototype` unless the `category` of the `System` is `ABSTRACT_PLATFORM_SYSTEM_DESCRIPTION`.⌋

[constr_6810] Applicable `category`s for data types in an abstract platform

Status: DRAFT
Imposition time: `IT_Apsd`

⌈

Category	Applicable to ...							Description
	`ApplicationDataType`	`ApplicationDeferredDataType`	`ApplicationArrayDataType`	`ApplicationRecordDataType`	`ApplicationPrimitiveDataType`	`ApplicationRecordElement`	`ApplicationArrayElement`	
`VALUE`					x	x	x	Contains a single value.
`STRUCTURE`				x		x	x	Holds one or several further elements which can have different `AutosarDataType`s.
`STRING`					x	x	x	Contains a single value interpreted as a text string (note that it appears as a single value for the application domain).
`ARRAY`			x			x	x	A fixed-sized array of sub-elements of the same type.
`BOOLEAN`					x	x	x	Contains a single boolean (true/false) state.

⌋

[constr_6812] Supported SwDataDefProps applicable to ApplicationDataTypes exclusive to the abstract platform

Status: DRAFT
Imposition time: IT_Apsd

⌈

Attributes of SwDataDefProps	Root Elem.				Attribute Existence per Category				
	ApplicationDataType	ApplicationDeferredDataType	ApplicationRecordElement	ApplicationArrayElement	VALUE	STRUCTURE	ARRAY	STRING	BOOLEAN
annotation	x	x	x	x	*	*	*	*	*
compuMethod	x				0..1				0..1
dataConstr.dataConstrRule.physConstrs	x		x	x	0..1		0..1		0..1
dataConstr.dataConstrRule.internalConstrs	x		x	x	d/c[1]		d/c		d/c
displayFormat	x		x	x	0..1		0..1	0..1	0..1
invalidValue	x				0..1			0..1	0..1
swTextProps	x							1	
unit	x				0..1			0..1	0..1
Other Attributes below the Root Element									
element:ApplicationRecordElement	x		x	x		1..*			
element:ApplicationArrayElement	x		x	x			1		
ApplicationArrayElement.arraySizeSemantics	x						0..1		
ApplicationArrayElement.maxNumberOfElements	x						1		

⌋

[constr_6814] Restriction of ApplicationDeferredDataType.category

Status: DRAFT
Imposition time: IT_Apsd

⌈The category of an ApplicationDeferredDataType shall be unassigned/undefined.⌋

2.2 FO_TPS_FeatureModelExchangeFormat

[constr_3657] Multiplicity of FMAttributeDef.max and FMAttributeDef.min

Imposition time: IT_FeatMod

⌈For each FMAttributeDef the attributes max and min shall exist.⌋

[1] don't care

[constr_3658] Multiplicity of FMFeatureDecomposition.category

Imposition time: IT_FeatMod

⌈For each FMFeatureDecomposition the attribute category shall exist.⌋

[constr_3659] Multiplicity of FMFeatureDecomposition.feature

Imposition time: IT_FeatMod

⌈For each FMFeatureDecomposition at least one reference in the role feature shall exist.⌋

[constr_3660] Multiplicity of FMFeatureRelation.feature

Imposition time: IT_FeatMod

⌈For each FMFeatureRelation at least one reference in the role feature shall exist.⌋

[constr_3661] Multiplicity of FMFeatureSelection.feature

Imposition time: IT_FeatMod

⌈For each FMFeatureSelection the reference in the role feature shall exist.⌋

[constr_3662] Multiplicity of FMFeatureSelection.state

Imposition time: IT_FeatMod

⌈For each FMFeatureSelection the attribute state shall exist.⌋

[constr_3663] Multiplicity of FMAttributeValue.definition

Imposition time: IT_FeatMod

⌈For each FMAttributeValue the reference in the role definition shall exist.⌋

[constr_3664] Multiplicity of FMAttributeValue.value

Imposition time: IT_FeatMod

⌈For each FMAttributeValue the attribute value shall exist.⌋

[constr_3665] Multiplicity of FMFormulaByFeaturesAndAttributes.attribute

Imposition time: IT_FeatMod

⌈For each FMFormulaByFeaturesAndAttributes the reference in the role attribute shall exist.⌋

[constr_3666] Multiplicity of FMFormulaByFeaturesAndAttributes.feature

Imposition time: IT_FeatMod

⌈For each FMFormulaByFeaturesAndAttributes the reference in the role feature shall exist.⌋

[constr_3667] Multiplicity of `FMFormulaByFeaturesAndSwSystemconsts.feature`

Imposition time: IT_FeatMod

⌈For each FMFormulaByFeaturesAndSwSystemconsts the reference in the role feature shall exist.⌋

[constr_5001] `FMFeatureRelation` shall not establish self-references

Imposition time: IT_FeatMod

⌈A FMFeatureRelation that is aggregated by a FMFeature f shall not reference f in the role feature. In other words: self-references are not allowed.⌋

[constr_5002] `FMFeatureSelectionSet` shall not have cycles in the `include` relation

Imposition time: IT_FeatMod

⌈Let S be a FMFeatureSelectionSet and let G be the *inclusion graph* for all FMFeatureSelectionSets as defined in [TPS_FMDT_00032]. There shall be no cycles in the inclusion graph.⌋

[constr_5003] `FMFeatureSelectionSet` shall not overwrite the state of included features

Imposition time: IT_FeatMod

⌈Let S be a FMFeatureSelectionSet that aggregates a FMFeatureSelection that has the state s and which refers to a FMFeature f in the role feature. Furthermore, let S_1 be a FMFeatureSelectionSet that aggregates a FMFeatureSelection that has the state s_1 and refers *to the same FMFeature* f in the role feature. Finally assume that S refers to S_1 in the role include.

Then the following conditions shall hold:

1. If the value of the attribute state of s_1 is undecided, then the value of the attribute state of s may be one of selected, deselected, and undecided.
2. If the value of the attribute state of s_1 is selected or deselected, then the value of the attribute state of s shall be the same as the attribute state in s_1, or undecided.
3. Any other constellation is considered an error.

⌋

[constr_5005] `FMFeature` shall not be referenced from more than one `FMFeatureDecomposition`

Imposition time: IT_FeatMod

⌈Let f be a FMFeature that is referenced from a FMFeatureDecomposition in the role feature. Then no other FMFeatureDecomposition shall reference f in the role feature.⌋

[constr_5007] `FMFeature` shall only be referenced from one `FMFeatureModel` in the role `feature`

Imposition time: `IT_FeatMod`

⌈Let f be a `FMFeature`, and F,F' be `FMFeatureModel`s where F references f in the role `feature`, and F' also references f in the role `feature`. Then $F = F'$.⌋

[constr_5008] If present, the root feature shall be part of the feature model

Imposition time: `IT_FeatMod`

⌈Let r be the `FMFeature` referenced from `FMFeatureModel` in the role `root`, and $\{f_1, f_2, \ldots, f_n\}$ the set of features referenced from the same `FMFeatureModel` in the role `feature`.

Then the following condition shall hold: $r \in \{f_1, f_2, \ldots, f_n\}$.⌋

[constr_5009] Root feature shall be present if and only if the feature model is not empty

Imposition time: `IT_FeatMod`

⌈If a `FMFeatureModel` refers to one or more `FMFeature` elements in the role `feature`, then exactly one of them shall be referenced by `FMFeatureModel` in the role `root`.

On the contrary, if `FMFeatureModel` does not refer to any `FMFeature`s in the role `feature`, then `root` shall be empty.⌋

[constr_5010] `FMFeatureDecomposition` may refer to a root feature of another feature model, but only once.

Imposition time: `IT_FeatMod`

⌈Let f_A be a `FMFeature` that is referenced by `FMFeatureModel` A in the role `feature`, but is also referenced from a `FMFeatureDecomposition` that is aggregated by a `FMFeature` f_B in the role `decomposition`.

Furthermore, let B be the `FMFeatureModel` that references f_B in the role `feature` with $A \neq B$. That is, f_A and f_B belong to different feature models.

Then *both* the following conditions shall hold:

1. f_A is referenced from A in the role `root`.
2. There is no other `FMFeatureDecomposition` (neither in B nor in any other `FMFeatureModel`) that references f_B in the role `feature`.

⌋

[constr_5011] `FMFormulaByFeaturesAndAttributes` can refer to `FMFeature`s and `FMAttributeDef`s, but not to system constants

Imposition time: IT_FeatMod

⌈A formula of class FMFormulaByFeaturesAndAttributes is an expression that can use FMFeatures and FMAttributeDefs, but is not allowed to use SwSystemconsts.⌋

[constr_5013] Attributes `min` and `max` of `FMFeatureDecomposition` reserved for `category` MULTIPLEFEATURE

Imposition time: IT_FeatMod

⌈The optional attributes min and max of FMFeatureDecomposition are only allowed to be present if the category of the FMFeatureDecomposition is MULTIPLEFEATURE.⌋

[constr_5018] `FMFeatureSelectionSet` shall not include the same feature twice

Imposition time: IT_FeatMod

⌈Let $\{s_1, s_2, \ldots, s_n\}$ be the set of FMFeatureSelection elements that are aggregated by a FMFeatureSelectionSet in the role selection. Furthermore, for each s_i, let f_i be the FMFeature that is referred to in the role feature. Then the following condition shall hold true:

$$\forall i, j \in \{1, 2, \ldots, n\} : i \neq j \Rightarrow f_i \neq f_j$$

⌋

[constr_5019] `FMFeatureModel` shall not contain the same `FMFeature` twice

Imposition time: IT_FeatMod

⌈Let F be a FMFeatureModel, and let f, f' be FMFeatures that are referenced from F in the role feature. Then $f \neq f'$.⌋

[constr_5020] Every `FMFeature` shall be contained in a `FMFeatureModel`

Imposition time: IT_FeatMod

⌈For every FMFeature f, there shall be a FMFeatureModel that refers to f in the role feature.⌋

[constr_5021] The underlying graph of a feature model shall be a tree.

Imposition time: IT_FeatMod

⌈Let F be a FMFeatureModel and G be the underlying graph of F as defined in [TPS_FMDT_00034]. Then G shall be a tree. Hence, we also refer to G as the *underlying tree* of F.⌋

[constr_5022] The root feature of a `FMFeatureModel` refers to the root of the underlying tree.

Imposition time: `IT_FeatMod`

⌈Let F be a `FMFeatureModel` and G be the underlying tree of F as defined in [TPS_FMDT_00034]. Furthermore, let r be the `FMFeature` referred to by the `root` feature of the `FMFeatureModel`.

Then the node in G which corresponds to r is the root of the tree G.⌋

[constr_5023] `FMFeatureSelectionSet` may only refer to `FMFeature`s from the associated `FMFeatureModel`

Imposition time: `IT_FeatMod`

⌈Let S be a `FMFeatureSelectionSet`, and $\{f_1, f_2, \ldots, f_n\}$ be its *feature set* ([TPS_FMDT_00009]). Furthermore, let $\{g_1, g_2, \ldots, g_m\}$ be the combined *feature sets* of the `FMFeatureModel`s to which S refers to in the role `featureModel`.

Then the following condition shall hold: $\{f_1, f_2, \ldots, f_n\} \subseteq \{g_1, g_2, \ldots, g_m\}$.⌋

[constr_5024] `FMFeatureSelectionSet` shall not include itself

Imposition time: `IT_FeatMod`

⌈Let S be a `FMFeatureSelectionSet` and let S' be the `FMFeatureSelectionSet` to which S refers to in the role `include`.

Then the following condition shall hold: $S \neq S'$.z⌋

[constr_5025] `FMFeatureSelectionSet` shall not overwrite the state of included features

Imposition time: `IT_FeatMod`

⌈Let S be a `FMFeatureSelectionSet` that aggregates a `FMFeatureSelection` that has the `state` s and which refers to a `FMFeature` f in the role `feature`. Furthermore, let S_1 (S_2) be a `FMFeatureSelectionSet` that aggregates a `FMFeatureSelection` that has the `state` s_1 (s_2) and refers *to the same* `FMFeature` f in the role `feature`. Finally assume that S refers to S_1 *and* S_2 in the role `include`.

Then the following conditions shall hold:

1. If the values of the attributes `state` of s_1 and s_2 are both `undecided`, then the value of the attribute `state` of s may be `selected`, `deselected` or `undecided`.
2. If the value of the attribute `state` of s_1 is `undecided` and the value of the attribute `state` of s_2 is `selected` or `deselected`, then the value of the attribute `state` of s shall be the same as the attribute `state` in s_2, or `undecided`.
3. If the value of the attribute `state` of s_2 is `undecided` and the value of the attribute `state` of s_1 is `selected` or `deselected`, then the value of the attribute `state` of s shall be the same as the attribute `state` in s_1, or `undecided`.

4. If the values of the attributes `state` of s_1 and s_2 are both either `selected` or `deselected`, then the value of the attribute `state` of s shall be the same as in attribute s_1, or `undecided`.
5. Any other constellation is considered an error.

⌋

[constr_5026] Semantics of attributes `max` and `min` in class `FMAttributeDef`

Imposition time: `IT_FeatMod`

⌈The following conditions shall hold for all instances of the class `FMAttributeDef`:

- `min` $\leq$ `defaultValue` $\leq$ `max` (`min` and `max` are both closed intervals)
- `min` $<$ `defaultValue` $\leq$ `max` (`min` is an open interval, `max` is a closed interval)
- `min` $<$ `defaultValue` $<$ `max` (`min` and `max` are both open intervals)
- `min` $\leq$ `defaultValue` $<$ `max` (`min` is a closed interval, `max` is an open interval)

⌋

[constr_5027] Semantics of attributes `max` and `min` of `FMAttributeDef` in class `FMAttributeValue`

Imposition time: `IT_FeatMod`

⌈Let v be the attribute `value` of an `FMAttributeValue` V that refers to `FMAttributeDef` D in the role `definition`. Furthermore, let min and max be the values of the attributes `min` and `max` of D.

The following condition shall hold true:

$$min \leq v \leq max$$

⌋

[constr_5028] Only one `FMAttributeValue` per `FMAttributeDef`

Imposition time: `IT_FeatMod`

⌈Let S be a `FMFeatureSelectionSet` whose `FMFeatureSelection`s aggregate `FMAttributeValue`s $\{v_1, v_2, \ldots, v_n\}$ in the role `attributeValue`. For each v_i, let f_i be the `FMFeature` to which v_i refers to in the role `attributeDef`. Then the following condition shall hold:

$$\forall i \in \{1, \ldots, n\} : i \neq j \Rightarrow f_i \neq f_j$$

⌋

2.3 FO_TPS_LogAndTrace

[constr_5098] Allowed SwDataDefProps attributes for DltArgumentProps.networkRep

Imposition time: IT_LogTrace

⌈

Attributes of SwDataDefProps	networkRep
annotation	N/A
baseType	D
compuMethod	D
dataConstr	D
displayFormat	D
displayPresentation	N/A
invalidValue	N/A
swComparisonVariable	N/A
swHostVariable	N/A
swTextProps	D
unit	D

⌋

[constr_5294] Existence of DltEcu.ecuId

Imposition time: IT_LogTrace

⌈For each DltEcu, the attribute ecuId shall exist.⌋

[constr_5295] Existence of DltApplication.context

Imposition time: IT_LogTrace

⌈Each DltApplication shall reference at least one DltContext in the role context.⌋

[constr_5296] Existence of DltApplication.applicationId

Imposition time: IT_LogTrace

⌈For each DltApplication, the attribute applicationId shall exist.⌋

[constr_5297] Existence of DltApplication.applicationDescription

Imposition time: IT_LogTrace

⌈For each DltApplication, the attribute applicationDescription shall exist.⌋

[constr_5298] Existence of DltContext.contextId

Imposition time: IT_LogTrace

⌈For each DltContext, the attribute contextId shall exist.⌋

[constr_5299] Existence of DltContext.contextDescription

Imposition time: IT_LogTrace

⌈For each DltContext, the attribute contextDescription shall exist.⌋

[constr_5300] Existence of `DltContext.dltMessage`

Imposition time: `IT_LogTrace`

⌈Each `DltContext` shall reference at least one `DltMessage` in the role `dltMessage`.⌋

[constr_5301] Existence of `DltMessage.messageId`

Imposition time: `IT_LogTrace`

⌈For each `DltMessage`, the attribute `messageId` shall exist.⌋

[constr_5303] Restriction of `baseTypeSize` of a `DltArgument`

Imposition time: `IT_LogTrace`

⌈The `baseTypeSize` in the `networkRep` of a `DltArgument` is restricted to 8, 16, 32, and 64 Bits.⌋

[constr_5304] Datatype of an Array

Imposition time: `IT_LogTrace`

⌈The `dltArgumentEntry` that is aggregated by `DltArgumentProps` that has the `length` attribute set to a value (represents an Array) shall not define a `SwBaseType` in the `networkRep` since the data type of the Array is described by the `SwBaseType` in the `networkRep` of the aggregating `DltArgumentProps`.⌋

[constr_5305] `CompuMethod` in `DltArgumentProps.networkRep`

Imposition time: `IT_LogTrace`

⌈The `CompuMethod` that is used in the `networkRep` of `DltArgumentProps` referenced by `DltArgument` is limited to `category` TEXTTABLE.⌋

[constr_5340] Range of `DltMessage.privacyLevel.privacyLevel`

Imposition time: `IT_LogTrace`

⌈The value of `DltMessage.privacyLevel.privacyLevel` shall be in the range between 0 and 255.⌋

[constr_5341] Range of `PrivacyLevel.compuMethod`

Imposition time: `IT_LogTrace`

⌈The `CompuMethod` that is referenced from `PrivacyLevel` in the role `compuMethod` shall have the category TEXTTABLE.⌋

[constr_5363] Allowed usage of attributes for description of payload data types

Imposition time: IT_LogTrace

⌈

Type	length	dltArgumentEntry	SwBaseType of top level Dlt ArgumentProps
Predefined Text	NA	NA	NA
primitive Type	NA	NA	D
String	D	NA	D
1-dimensional Array	D	NA	D
n-dimensional Array	D	D	D
Struct	NA	D	NA

⌋

[constr_5364] Allowed usage of attributes in case of a dltArgumentEntry

Imposition time: IT_LogTrace

⌈

dltArgumentEntry type	length	dltArgumentEntry	SwBaseType of DltArgumentEntry
Struct member	NA	D	D
Array dimension	D	D	NA

⌋

[constr_9364] Mutually exclusive existence of DltArgument.optional and DltArgument.predefinedText attributes

Imposition time: IT_LogTrace

⌈A DltArgument element shall never define the attributes DltArgument.optional and DltArgument.predefinedText at the same time.⌋

[constr_9365] DltArgument with predefinedText

Imposition time: IT_LogTrace

⌈If the predefinedText of a DltArgument is set to true then the DltArgument shall not reference DltArgumentProps in the role dltArgumentProps.⌋

[constr_9405] DltArguments with variableLength or DltArgument.optional settings inside of a DltMessage

Imposition time: IT_LogTrace

⌈For each DltMessage, there shall be at maximum only one DltArgument which sets the attribute DltArgumentProps.variableLength or DltArgument.optional to 'true'.⌋

[constr_9406] Setting of `DltArgumentProps.variableLength` to true

Imposition time: `IT_LogTrace`

⌈`DltArgumentProps.variableLength` is only allowed to be set to 'true' if the `DltArgument` represents a String (according to [TPS_DLTXT_00007]) or an Array (according to [TPS_DLTXT_00011] or [TPS_DLTXT_00012]).⌋

2.4 FO_TPS_SecurityExtractTemplate

[constr_5600] Valid interval for attribute `SecurityEventDefinition.id` ⌈The valid interval for attribute `SecurityEventDefinition.id` is 0..65534.⌋

[constr_5601] Uniqueness of `SecurityEventDefinition.id` ⌈Within the scope of an IDS, i.e. for all `SecurityEventDefinition`s referenced by the same `IdsDesign`, there shall be no attribute `id` of any other `SecurityEventDefinition` that has the same value.⌋

[constr_5602] Valid interval for attribute `SecurityEventOneEveryNFilter.n` ⌈The valid interval for attribute `SecurityEventOneEveryNFilter.n` is 2..65535.⌋

[constr_5603] Valid interval for attribute `SecurityEventAggregationFilter.minimumIntervalLength` ⌈The valid interval for attribute `SecurityEventAggregationFilter.minimumIntervalLength` is]0..INF[seconds.⌋

[constr_5604] Valid interval for attribute `SecurityEventThresholdFilter.intervalLength` ⌈The valid interval for attribute `SecurityEventThresholdFilter.intervalLength` is]0..INF[seconds.⌋

[constr_5605] Valid interval for attribute `SecurityEventThresholdFilter.thresholdNumber` ⌈The valid interval for attribute `SecurityEventThresholdFilter.thresholdNumber` is 2..65535.⌋

[constr_5606] Valid interval for attribute `IdsmRateLimitation.timeInterval` ⌈The valid interval for attribute `IdsmRateLimitation.timeInterval` is 0..65535 seconds.⌋

[constr_5607] Valid interval for attribute `IdsmRateLimitation.maxEventsInInterval` ⌈The valid interval for attribute `IdsmRateLimitation.maxEventsInInterval` is 1..65535.⌋

[constr_5608] Valid interval for attribute `IdsmTrafficLimitation.timeInterval` ⌈The valid interval for attribute `IdsmTrafficLimitation.timeInterval` is 0..65535 seconds.⌋

[constr_5609] Valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval ⌈The valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval is 1..65535.⌋

[constr_5610] Unambiguous definition of execution platform for an IdsmInstance ⌈For the meta-class IdsmInstance, either the reference in the role ecuInstance or the reference in the role idsmInstance shall be defined in order to ensure that the platform (Classic or Adaptive) on which an IdsmInstance is targeted to run is unambiguously defined.⌋

[constr_5612] Unambiguous definition of platform-dependent network configuration for an IdsmInstance ⌈For the meta-class IdsmInstance, either the configuration of one GeneralPurposeIPdu with category="IDS" (for the Classic Platform as specified in [TPS_SECXT_01038]) or the network configuration through the reference idsmInstance (for the Adaptive Platform as specified in [TPS_SECXT_01039]) shall be defined in order to ensure that the platform-dependent network configuration is unambiguously defined.⌋

[constr_5613] Unambiguous definition of SecurityEventStateFilter for CP or AP ⌈For SecurityEventStateFilter, either the references in the role blockIfStateActiveCp or the references in the role blockIfStateActiveAp shall be defined in order to ensure the unambiguous applicability of the SecurityEventStateFilter towards the Classic or the Adaptive Platform.⌋

[constr_5614] Upper bound for multiplicity of BlockStates aggregated by IdsmInstance ⌈For the meta-class IdsmInstance, the maximum number of aggregated BlockStates in the role blockState shall be 16.⌋

[constr_5615] Restriction of SecurityEventStateFilter referencing BlockStates on CP ⌈For a SecurityEventStateFilter on the Classic Platform, the references in the role blockIfStateActiveCp shall only reference those BlockStates that are aggregated in the role blockState by the IdsmInstance which is mapped (by SecurityEventContextMapping) to that SecurityEventFilterChain of which the SecurityEventStateFilter is part of.⌋

[constr_9339] SecurityEventContextDataElement.maxLength usage restriction ⌈The SecurityEventContextDataElement.maxLength attribute shall only be set if the SecurityEventContextDataElement.networkRepresentation defines an Array or a String.⌋

[constr_9340] Datatype of an Array ⌈The nestedContextData that is aggregated by a SecurityEventContextDataElement that has the maxLength attribute set to a value (represents an Array) shall not define a SwBaseType in the networkRepresentation since the data type of the Array is described by the SwBaseType in the networkRepresentation of the aggregating SecurityEventContextDataElement.⌋

[constr_9341] CompuMethod in SecurityEventContextDataElement.networkRepresentation ⌈The `CompuMethod` that is used in the `networkRepresentation` of a `SecurityEventContextDataElement` is limited to category TEXTTABLE.⌋

[constr_9342] Allowed range of `SecurityEventContextDataDefinition.version` ⌈The value given for `SecurityEventContextDataDefinition.version` shall be in the range from 0 to 65535.⌋

[constr_12000] Usage of references in the context of `SecurityEventReportInstanceValue` ⌈Within each `SecurityEventReportInstanceValue`, exactly one of the references in the role

- `flatObject`
- `object`

shall exist.⌋

[constr_12001] Existence of attribute `SecurityEventReportInstanceValue.id` ⌈For each `SecurityEventReportInstanceValue`, attribute `id` shall exist.⌋

[constr_12002] Existence of attribute `SecurityEventReportInstanceDefinition.targetType` ⌈For each `SecurityEventReportInstanceDefinition`, the attribute `targetType` shall exist.⌋

[constr_12003] Existence of reference `SecurityEventReportInstanceValue.contextDataElementIdentification` ⌈In the context of a `SecurityEventContextDataElement`, if the reference in the role `SecurityEventContextDataElement.securityEventReportInstanceDefinition` does not exist, then the `SecurityEventContextDataElement` shall not be referenced in the role `SecurityEventReportInstanceValue.contextDataElementIdentification`.⌋

[constr_12004] Valid interval for attribute `SecurityEventContextProps.sensorInstanceId` ⌈The valid interval for attribute `SecurityEventContextProps.sensorInstanceId` is 0..63.⌋

[constr_12005] Valid interval for attribute `IdsmInstance.idsmInstanceId` ⌈The valid interval for attribute `IdsmInstance.idsmInstanceId` is 0..1023.⌋

[constr_12006] Existence of attribute `IdsmInstance.idsmInstanceId` ⌈For each `IdsmInstance`, the attribute `idsmInstanceId` shall exist.⌋

[constr_12007] Existence of attribute `IdsmInstance.rateLimitationFilter` ⌈For each `IdsmInstance`, the reference to `IdsmRateLimitation` in the role `rateLimitationFilter` shall exist.⌋

[constr_12008] Existence of attribute IdsmInstance.trafficLimitationFilter ⌈For each IdsmInstance, the reference to IdsmTrafficLimitation in the role trafficLimitationFilter shall exist.⌋

[constr_12009] Existence of attribute IdsmTrafficLimitation.maxBytesInInterval ⌈For each IdsmTrafficLimitation, the attribute maxBytesInInterval shall exist.⌋

[constr_12010] Existence of attribute IdsmTrafficLimitation.timeInterval ⌈For each IdsmTrafficLimitation, the attribute timeInterval shall exist.⌋

[constr_12011] Existence of attribute SecurityEventDefinition.id ⌈For each SecurityEventDefinition, the attribute id shall exist.⌋

[constr_12012] Existence of attribute SecurityEventAggregationFilter.contextDataSource ⌈For each SecurityEventAggregationFilter, the attribute contextDataSource shall exist.⌋

[constr_12013] Existence of attribute SecurityEventAggregationFilter.minimumIntervalLength ⌈For each SecurityEventAggregationFilter, the attribute minimumIntervalLength shall exist.⌋

[constr_12014] Existence of attribute SecurityEventOneEveryNFilter.n ⌈For each SecurityEventOneEveryNFilter, the attribute n shall exist.⌋

[constr_12015] Existence of attribute SecurityEventThresholdFilter.intervalLength ⌈For each SecurityEventThresholdFilter, the attribute intervalLength shall exist.⌋

[constr_12016] Existence of attribute SecurityEventThresholdFilter.thresholdNumber ⌈For each SecurityEventThresholdFilter, the attribute thresholdNumber shall exist.⌋

[constr_12017] Existence of attribute SecurityEventContextMapping.idsmInstance ⌈For each SecurityEventContextMapping, the reference to IdsmInstance in the role idsmInstance shall exist.⌋

[constr_12018] Existence of attribute SecurityEventContextMapping.mappedSecurityEvent ⌈For each SecurityEventContextMapping, the aggregation of SecurityEventContextProps in the role mappedSecurityEvent shall exist.⌋

[constr_12019] Existence of attribute SecurityEventContextProps.defaultReportingMode ⌈For each SecurityEventContextProps, the attribute defaultReportingMode shall exist.⌋

[constr_12020] Existence of attribute SecurityEventContextProps.securityEvent ⌈For each SecurityEventContextProps, the reference to SecurityEventDefinition in the role securityEvent shall exist.⌋

[constr_12021] Existence of attribute SecurityEventContextProps.sensorInstanceId ⌈For each SecurityEventContextProps, the attribute sensorInstanceId shall exist.⌋

[constr_12022] Existence of role IdsmRateLimitation.maxEventsInInterval

Imposition time: IT_SecEx

⌈For each IdsmRateLimitation, the reference in the role maxEventsInInterval shall exist.⌋

[constr_12023] Existence of role IdsmRateLimitation.timeInterval

Imposition time: IT_SecEx

⌈For each IdsmRateLimitation, the reference in the role timeInterval shall exist.⌋

[constr_12024] Existence of role SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster

Imposition time: IT_SecEx

⌈For each SecurityEventContextMappingFunctionalCluster, the reference in the role affectedFunctionalCluster shall exist.⌋

[constr_12025] Existence of role SecurityEventContextMappingApplication.affectedApplication

Imposition time: IT_SecEx

⌈For each SecurityEventContextMappingApplication, the reference in the role affectedApplication shall exist.⌋

A Mentioned Class Tables

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*, *AtpClassifier*, *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.
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.1: 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.2: ApplicationArrayElement

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.

▽

△

Class	***ApplicationDataType*** (abstract)			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*ApplicationCompositeDataType*, ApplicationDeferredDataType, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.3: ApplicationDataType

Class	**ApplicationDeferredDataType**			
Note	A placeholder data type in which the precise application data type is deferred to a later stage. **Tags:** atp.Status=draft atp.recommendedPackage=ApplicationDataTypes This Class is only used by the AUTOSAR Foundation.			
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.4: ApplicationDeferredDataType

Class	**ApplicationInterface**			
Note	This represents the ability to define a PortInterface that consists of a composition of commands (method calls), indications (events) and attributes (fields) **Tags:** atp.Status=draft atp.recommendedPackage=Interfaces This Class is only used by the AUTOSAR Foundation.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
attribute	Field	*	aggr	This represents the set of attributes defined in the context of an Abstract Platform ApplicationInterface. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=attribute, attribute.shortName, attribute.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=blueprintDerivationTime
command	ClientServerOperation	*	aggr	This represents the collection of commands or function calls (with optional data arguments) defined in the context of an ApplicationInterface. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=command, command.shortName, command.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=blueprintDerivationTime

▽

△

Class	**ApplicationInterface**			
indication	VariableDataPrototype	*	aggr	This represents the collection of indication or events (with optional data argument) defined in the context of an ApplicationInterface. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=indication, indication.shortName, indication.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=blueprintDerivationTime

Table A.5: ApplicationInterface

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.6: 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.7: 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*

▽

△

Class	**ApplicationRecordElement**			
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.8: ApplicationRecordElement

Class	***AutosarDataType*** (abstract)			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *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.9: AutosarDataType

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

▽

△

Class	**BaseTypeDirectDefinition**			
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.10: BaseTypeDirectDefinition

Class	**BlockState**			
Note	This meta-class defines a block state that is part of the collection of block states belonging to a specific IdsmInstance. The IdsM shall discard any reported security event that is mapped to a filter chain containing a SecurityEventStateFilter that references the block state which is currently active in the IdsM. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmInstance.blockState			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.11: BlockState

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*
component	SwComponent Prototype	*	aggr	**Stereotypes:** atpVariation **Tags:** atp.Splitkey=component, component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**CompositionSwComponentType**			
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
physical Dimension Mapping	PhysicalDimension MappingSet	0..1	ref	This reference identifies the PhysicalDimensionMappingSet that is applicable in the context of the enclosing CompositionSwComponentType. The PhysicalDimensionMappings contained in the PhysicalDimensionMappingSet shall be taken into account for the assessment of the compatibility of PhysicalDimensions in the context of creation of a PortInterfaceMapping in the scope of the CompositionSwComponentType. **Tags:** atp.Splitkey=physicalDimensionMapping

Table A.12: CompositionSwComponentType

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			
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

▽

△

Class	**CompuMethod**			
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.13: CompuMethod

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.14: DataConstr

Class	**DataConstrRule**			
Note	This meta-class represents the ability to express one specific data constraint rule.			
Base	*ARObject*			
Aggregated by	DataConstr.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.15: DataConstrRule

Class	**DltApplication**			
Note	This meta-class represents the application from which the log and trace message originates.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltEcu.application			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Description	String	0..1	attr	This attribute can be used to describe the applicationId that is used in the log and trace message in more detail.
applicationId	String	0..1	attr	This attribute identifies the SW-C/BSW module in the log and trace message.
context	DltContext	*	ref	Definition of ContextIds for the Application. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=context, context.dltContext, context.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table A.16: DltApplication

Class	**DltArgument**			
Note	This element defines an Argument in a DltMessage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltMessage.dltArgument			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dltArgument Props	DltArgumentProps	0..1	ref	Additional properties of DltArguments **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dltArgumentProps, dltArgumentProps.dlt ArgumentProps, dltArgumentProps.variationPoint.short Label vh.latestBindingTime=systemDesignTime
optional	Boolean	0..1	attr	This attribute defines whether the argument is optional or not. If set to true, the argument can be omitted from the payload of a DLT message.
predefinedText	Boolean	0..1	attr	This attribute defines whether the DltArgument is a predefinedText (Static Data).

Table A.17: DltArgument

Class	**DltArgumentProps**			
Note	This element defines reusable DltArgument properties that may be used by DltArguments of different Dlt Messages.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltArgumentProps.dltArgumentEntry, DltArgumentPropsSet.dltArgumentProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dltArgument Entry	DltArgumentProps	*	aggr	This aggregation is used to describe subElements of Dlt ArgumentProps that defines a Structure. **Tags:** atp.Splitkey=dltArgumentEntry, dltArgument Entry.shortName
length	PositiveInteger	0..1	attr	Describes the DltArgument length in case of Arrays and Strings in number of BaseTypes.

▽

△

Class	**DltArgumentProps**			
networkRep	SwDataDefProps	0..1	aggr	Definition of the networkRepresentation of the Dlt Argument. **Tags:** atp.Splitkey=networkRep
variableLength	Boolean	0..1	attr	This attribute defines whether the length of the Dlt Argument is variable (determined at runtime) or not.

Table A.18: DltArgumentProps

Class	**DltContext**			
Note	This meta-class represents the Context that groups Log and Trace messages that are generated by an application. **Tags:** atp.recommendedPackage=DltContexts			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context Description	String	0..1	attr	This attribute can be used to describe the contextId that is used in the log and trace message in more detail.
contextId	String	0..1	attr	This attribute is used to group log and trace messages produced by an application to distinguish functionality.
dltMessage	DltMessage	*	ref	Group of Log and Trace messages assigned to the Dlt Context **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dltMessage, dltMessage.dltMessage, dlt Message.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table A.19: DltContext

Class	**DltEcu**			
Note	This element represents an Ecu or Machine that produces logging and tracing information. **Tags:** atp.recommendedPackage=DltEcus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application	DltApplication	*	aggr	Application on DltEcu that provides log or trace data. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=application, application.shortName, application.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
ecuId	String	0..1	attr	This attribute defines the name of the ECU for use within the Dlt protocol.

Table A.20: DltEcu

Class	**DltMessage**			
Note	This element defines a DltMessage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LogAndTraceMessageCollectionSet.dltMessage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dltArgument (ordered)	DltArgument	*	aggr	Ordered collection of DltArguments in the DltMessage.
messageId	PositiveInteger	0..1	attr	This attribute defines the unique Id for the DltMessage.
messageLine Number	PositiveInteger	0..1	attr	This attribute describes the position in the source file in which this log message was called.
messageSource File	String	0..1	attr	This attribute describes the source file in which this log message was called.
messageType Info	String	0..1	attr	This attribute describes the message Type
privacyLevel	PrivacyLevel	0..1	aggr	The Privacy Level helps to identify the Log and Trace content towards the degree of privacy to it. **Tags:** atp.Splitkey=privacyLevel

Table A.21: DltMessage

Class	**FMAttributeDef**			
Note	This metaclass represents the ability to define attributes for a feature.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FMFeature.attributeDef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultValue	Numerical	0..1	attr	This represents the default value of the attribute.
max	Limit	0..1	attr	Maximum possible value for the value of this attribute
min	Limit	0..1	attr	Minimum possible value for the value of this attribute

Table A.22: FMAttributeDef

Class	**FMAttributeValue**			
Note	This defines a value for the attribute that is referred to in the role definition.			
Base	*ARObject*			
Aggregated by	FMFeatureSelection.attributeValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
definition	FMAttributeDef	0..1	ref	This refers to the definition of this attribute. **Stereotypes:** atpIdentityContributor
value	Numerical	0..1	attr	This represents the value of this attribute.

Table A.23: FMAttributeValue

Class	**FMFeature**
Note	A FMFeature describes an essential characteristic of a product. Each FMFeature is contained in exactly one FMFeatureModel. **Tags:** atp.recommendedPackage=FMFeatureModels
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**FMFeature**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
attributeDef	FMAttributeDef	*	aggr	This defines the attributes of the given feature. **Tags:** atp.Splitkey=attributeDef, attributeDef.shortName
decomposition	FMFeature Decomposition	*	aggr	Lists the sub-features of a feature. **Tags:** atp.Splitkey=decomposition
maximum IntendedBinding Time	BindingTimeEnum	0..1	attr	Defines an upper bound for the binding time of the variation points that are associated with the FMFeature. This attribute is meant as a hint for the development process.
minimum IntendedBinding Time	BindingTimeEnum	0..1	attr	Defines a lower bound for the binding time of the variation points that are associated with the FMFeature. This attribute is meant as a hint for the development process.
relation	FMFeatureRelation	*	aggr	Defines relations for FMFeatures, for example dependencies on other FMFeatures, or conflicts with other FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if all its relations are fulfilled. **Tags:** atp.Splitkey=relation, relation.shortName
restriction	FMFeatureRestriction	*	aggr	Defines restrictions for FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if at least one of its restrictions evaluates to true. **Tags:** atp.Splitkey=restriction, restriction.shortName

Table A.24: FMFeature

Class	**FMFeatureDecomposition**			
Note	A FMFeatureDecomposition describes dependencies between a list of features and their parent feature (i.e., the FMFeature that aggregates the FMFeatureDecomposition). The kind of dependency is defined by the attribute category.			
Base	*ARObject*			
Aggregated by	FMFeature.decomposition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	CategoryString	0..1	attr	The category of a FMFeatureDecomposition defines the type of dependency that is defined by the FMFeature Decomposition. There are four different categories: MANDATORYFEATURE, OPTIONALFEATURE, ALTERNATIVEFEATURE, and MULTIPLEFEATURE.
feature	FMFeature	*	ref	The features that are affected by the dependency defined by the FMFeatureDecomposition. **Tags:** atp.Splitkey=feature
max	PositiveInteger	0..1	attr	For a dependency of category MULTIPLEFEATURE, this defines the maximum number of features allowed.
min	PositiveInteger	0..1	attr	For a dependency of category MULTIPLEFEATURE, this defines the minimum number of features allowed.

Table A.25: FMFeatureDecomposition

Class	**FMFeatureModel**
Note	A Feature model describes the features of a product line and their dependencies. Feature models are an optional part of an AUTOSAR model. **Tags:** atp.recommendedPackage=FMFeatureModels

▽

△

Class	**FMFeatureModel**			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
feature	FMFeature	*	ref	"feature" holds the list of features of the feature model. No FMFeature may be contained twice in this list. Also, each FMFeature may be contained on only one feature model. **Tags:** atp.Splitkey=feature
root	FMFeature	0..1	ref	The features of a feature model define a tree. The attribute root points to the root of this tree. **Tags:** atp.Splitkey=root

Table A.26: FMFeatureModel

Class	**FMFeatureRelation**			
Note	Defines relations for FMFeatures, for example dependencies on other FMFeatures, or conflicts with other FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if all its relations are fulfilled.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FMFeature.relation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
feature	FMFeature	*	ref	The FMFeature that is targeted by this FMFeature Relation. **Tags:** atp.Splitkey=feature
restriction	FMConditionByFeatures AndAttributes	0..1	aggr	If given, the condition shall evaluate to true, in order for the FMFeatureRelation to be active. **Tags:** atp.Splitkey=restriction

Table A.27: FMFeatureRelation

Class	**FMFeatureSelection**			
Note	A FMFeatureSelection represents the state of a particular FMFeature within a FMFeatureSelectionSet.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FMFeatureSelectionSet.selection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
attributeValue	FMAttributeValue	*	aggr	This defines a value for the attribute that is referred to in the role definition. Note that a FMFeatureSelection cannot include two FMAttributeValues that refer to the same FMAttributeDef in the role definition. **Tags:** atp.Splitkey=attributeValue, attributeValue.definition xml.sequenceOffset=50
feature	FMFeature	0..1	ref	The FMFeature whose state is defined by this FMFeature Selection. **Tags:** atp.Splitkey=feature xml.sequenceOffset=10

▽

△

Class	**FMFeatureSelection**			
maximum SelectedBinding Time	BindingTimeEnum	0..1	attr	Defines an upper bound for the binding time of the variation points that are associated with the FMFeature, and refines its maximumIntendedBindingTime. This attribute is meant as a hint for the development process. **Tags:** xml.sequenceOffset=40
minimum SelectedBinding Time	BindingTimeEnum	0..1	attr	Defines a lower bound for the binding time of the variation points that are associated with the FMFeature, and refines its minimumIntendedBindingTime. This attribute is meant as a hint for the development process. **Tags:** xml.sequenceOffset=30
state	FMFeatureSelection State	0..1	attr	Defines how the FMFeature that is described by this FMFeatureSelection contributes to the FMFeature SelectionSet. A FMFeature may have the state selected, deselected or undecided. **Tags:** xml.sequenceOffset=20

Table A.28: FMFeatureSelection

Class	**FMFeatureSelectionSet**			
Note	A FMFeatureSelectionSet is a set of FMFeatures that describes a specific product. **Tags:** atp.recommendedPackage=FMFeatureModelSelectionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
featureModel	FMFeatureModel	*	ref	All FMFeatures in this FMFeatureSelectionSet shall be part of the referenced FMFeatureModel. **Tags:** atp.Splitkey=featureModel
include	FMFeatureSelectionSet	*	ref	Each FMFeatureSelectionSet may include one or more FMFeatureSelectionSets. This establishes a hierarchy among FMFeatureSelectionSets. See constr_5003 and constr_5025 for details. **Tags:** atp.Splitkey=include
selection	FMFeatureSelection	*	aggr	The set of FMFeatureSelections of this FMFeature SelectionSet. **Tags:** atp.Splitkey=selection, selection.shortName

Table A.29: FMFeatureSelectionSet

Enumeration	**FMFeatureSelectionState**
Note	Defines how a particular FMFeature contributes to a FMFSelectionSet.
Aggregated by	FMFeatureSelection.state
Literal	*Description*
deselected	The feature is excluded from the selection. **Tags:** atp.EnumerationLiteralIndex=0
selected	The feature is included in the selection. **Tags:** atp.EnumerationLiteralIndex=1
undecided	It is not yet decided whether the feature shall be included into or excluded from the selection. **Tags:** atp.EnumerationLiteralIndex=2

Table A.30: FMFeatureSelectionState

Class	«atpMixedString» ***FMFormulaByFeaturesAndAttributes*** (abstract)			
Note	An expression that has the syntax of the AUTOSAR formula language but uses only references to features or feature attributes (not system constants) as operands.			
Base	*ARObject*, *FormulaExpression*			
Subclasses	FMConditionByFeaturesAndAttributes			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
attribute	FMAttributeDef	0..1	ref	An expression of type FMFormulaByFeaturesAnd Attributes may refer to attributes of FMFeatures.
feature	FMFeature	0..1	ref	An expression of type FMFormulaByFeaturesAnd Attributes may refer to FMFeatures.

Table A.31: FMFormulaByFeaturesAndAttributes

Class	«atpMixedString» ***FMFormulaByFeaturesAndSwSystemconsts*** (abstract)			
Note	An expression that has the syntax of the AUTOSAR formula language and may use references to features or system constants as operands.			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Subclasses	FMConditionByFeaturesAndSwSystemconsts			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
feature	FMFeature	0..1	ref	An expression of type FMFormulaByFeaturesAndSw Systemconsts may refer to FMFeatures.

Table A.32: FMFormulaByFeaturesAndSwSystemconsts

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.33: GeneralPurposeIPdu

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*, *AbstractFunctionalClusterDesign*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstance Filter*, *AbstractServiceInstance*, ApplicationEndpoint, ApplicationError, AppliedStandard, Artifact Checksum, *AtpBlueprint*, *AtpClassifier*, *AtpFeature*, AutosarOperationArgumentInstance, Autosar VariableInstance, BlockState, *BuildActionEntity*, BuildActionEnvironment, CanSecSecureChannel, Chapter, ClientIdDefinition, ClientServerOperation, Code, *CollectableElement*, ComManagement Mapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, *CouplingPortAbstractShaper*, *CouplingPort StructuralElement*, CryptoKeySlot, *CryptoServiceMapping*, DataPrototypeGroup, DataPrototype TransformationPropsIdent, DataTransformation, *DdsAbstractServiceInstanceElementCp*, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, DiagnosticExtendedDataRecordElement, DiagnosticFunctionInhibit Source, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, Dlt ArgumentProps, DltMessage, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, Ethernet WakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeature Relation, FMFeatureRestriction, FMFeatureSelection, FlexrayTpPduPool, *FrameTriggering*, GlobalTime Gateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttributeDef, HwAttributeLiteralDef, Hw Pin, HwPinGroup, *IEEE1722TpAcfBus*, *IEEE1722TpAcfBusPart*, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignalTriggering, *IdentCaption*, ImpositionTime, InternalTriggeringPoint, Keyword, LifeCycleState, Linker, MacAddressVlanMembership, MacMulticastGroup, MacSecKay Participant, McDataInstance, MemorySection, ModeDeclaration, ModeDeclarationMapping, ModeSwitch Point, ModeSwitchSenderComSpecProps, NetworkEndpoint, *NmCluster*, *NmNode*, *Packageable Element*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, Per InstanceMemory, *PhysicalChannel*, PortGroup, *PortInterfaceMapping*, QueuedReceiverComSpecProps, ResourceConsumption, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutable Entity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntity Group, *SdgAttribute*, SdgClass, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, ServerComSpecProps, *ServiceNeeds*, SignalServiceTranslationEventProps, SignalServiceTranslation Props, SocketAddress, SomeipTpChannel, *StackUsage*, StaticSocketConnection, StructuredReq, Sw GenericAxisParamType, SwServiceArg, SwcServiceDependency, SystemMapping, *TimeBaseResource*, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, Timing ExtensionResource, TimingModeInstance, Topic1, TpAddress, TraceableTable, TraceableText, *Traced Failure*, TransformationISignalPropsIdent, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, 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
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

▽

△

Class	***Identifiable*** (abstract)			
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.34: Identifiable

Class	**IdsDesign**			
Note	This meta-class represents the root element of a SecurityExtract file for IDS development. It defines the scope of an IDS to be designed and implemented by referencing all SecurityExtract meta-classes that need to be included into the IDS development process. **Tags:** atp.recommendedPackage=IdsDesigns			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**IdsDesign**			
element	IdsCommonElement	*	ref	This reference includes an element with IDS related definitions into the IdsDesign. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.idsCommonElement, element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table A.35: IdsDesign

Class	**IdsmInstance**			
Note	This meta-class provides the ability to create a relation between an EcuInstance and a specific class of filters for security events that apply for all security events reported on the referenced EcuInstance. **Tags:** atp.recommendedPackage=IdsmInstanceToEcuInstanceMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blockState	BlockState	*	aggr	This reference defines the BlockState in the collection BlockStateSet. **Tags:** atp.Splitkey=blockState, blockState.shortName This Attribute is only used by the AUTOSAR Classic Platform.
ecuInstance	EcuInstance	0..1	ref	This reference identifies the EcuInstance whose security events (of any type) shall be limited by the specific class of filters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuInstance, ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
idsmInstanceId	PositiveInteger	0..1	attr	This attribute is used to provide a source identification in the context of reporting security events..
rateLimitation Filter	IdsmRateLimitation	0..1	ref	This reference identifies the applicable rate limitation filter for all security events on the related EcuInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rateLimitationFilter, rateLimitationFilter.idsmRateLimitation, rateLimitationFilter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
signature SupportCp	IdsmSignatureSupportCp	0..1	aggr	The existence of this aggregation specifies that the IdsM shall add a signature to the QSEv messages it sends onto the network. The cryptographic algorithm and key to be used for this signature is further specified by the aggregated meta-class specifically for the Classic Platform. **Tags:** atp.Splitkey=signatureSupportCp This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**IdsmInstance**			
timestamp Format	String	0..1	attr	The existence of this attribute specifies that the IdsM shall add a timestamp to the QSEv messages it sends onto the network. I.e., if this attribute does not exist, no timestamp shall be added to the QSEv messages. The content of this attribute further specifies the timestamp format as follows: - "AUTOSAR" defines AUTOSAR standardized timestamp format according to the Synchronized Time-Base Manager - Any other string defines a proprietary timestamp format. Note: A string defining a proprietary timestamp format shall be prefixed by a company-specific name fragment to avoid collisions.
trafficLimitation Filter	IdsmTrafficLimitation	0..1	ref	This reference identifies the applicable traffic limitation filter for all security events on the related EcuInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=trafficLimitationFilter, trafficLimitation Filter.idsmTrafficLimitation, trafficLimitationFilter.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.36: IdsmInstance

Class	**IdsmModuleInstantiation**			
Note	This meta-class defines the attributes for the IdsM configuration on a specific machine. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *AdaptiveModuleInstantiation*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *IdsPlatformInstantiation*, *MultilanguageReferrable*, *NonOsModuleInstantiation*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, Machine.moduleInstantiation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
idsmInstance	IdsmInstance	0..1	ref	This reference identifies the IdsmInstance that is associated with the enclosing IdsmModuleInstantiation. **Tags:** atp.Splitkey=idsmInstance
queueSize	PositiveInteger	0..1	attr	Length of the queue in which IDS events can be stored before processing, if immediate processing is not possible.

Table A.37: IdsmModuleInstantiation

Class	**IdsmRateLimitation**			
Note	This meta-class represents the configuration of a rate limitation filter for security events. This means that security events are dropped if the number of events (of any type) processed within a configurable time window is greater than a configurable threshold.			
Base	*ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmProperties.rateLimitationFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxEventsIn Interval	PositiveInteger	0..1	attr	This attribute configures the threshold for dropping security events if the number of all processed security events exceeds the threshold in the respective time interval.

▽

△

Class	**IdsmRateLimitation**			
timeInterval	Float	0..1	attr	This attribute configures the length of the time interval in seconds for dropping security events if the number of all processed security events exceeds the configurable threshold within the respective time interval.

Table A.38: IdsmRateLimitation

Class	**IdsmTrafficLimitation**			
Note	This meta-class represents the configuration of a traffic limitation filter for Security Events. This means that security events are dropped if the size (in terms of bandwidth) of security events (of any type) processed within a configurable time window is greater than a configurable threshold.			
Base	*ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmProperties.trafficLimitationFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxBytesIn Interval	PositiveInteger	0..1	attr	This attribute configures the threshold for dropping security events if the size of all processed security events exceeds the threshold in the respective time interval.
timeInterval	Float	0..1	attr	This attribute configures the length of the time interval in seconds for dropping security events if the size of all processed security events exceeds the configurable threshold within the respective time interval.

Table A.39: IdsmTrafficLimitation

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***
–	–	–	–	–

Table A.40: PortPrototype

Class	**PrivacyLevel**			
Note	This meta-class defines the Privacy Level for a Log and Trace content.			
Base	*ARObject*			
Aggregated by	DltMessage.privacyLevel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuMethod	CompuMethod	0..1	ref	Reference to CompuMethod of category TEXTTABLE that defines the supported user-defined privacy levels. **Tags:** atp.Splitkey=compuMethod
privacyLevel	PositiveInteger	0..1	attr	The value that represents the privacy level and is transported in the Extension Header.

Table A.41: PrivacyLevel

Class	**SecurityEventAggregationFilter**			
Note	This meta-class represents the aggregation filter that aggregates all security events occurring within a configured time frame into one (i.e. the last reported) security event.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.aggregation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextData Source	SecurityEventContext DataSourceEnum	0..1	attr	This attributes defines whether the context data of the first or last time-aggregated security event shall be used for the resulting qualified security event.
minimum IntervalLength	TimeValue	0..1	attr	This attribute represents the configuration of the minimum time window in seconds for the aggregation filter.

Table A.42: SecurityEventAggregationFilter

Class	**SecurityEventContextDataDefinition**			
Note	This meta-class represents the possibility to add context data to the referencing SecurityEventDefinition. **Tags:** atp.recommendedPackage=SecurityEventContextDataDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextData Element (ordered)	SecurityEventContext DataElement	*	aggr	Description of contained SecurityEventContextData Elements. If the SecurityEventContextDataDefintion has a primitive type then only one SecurityEventContextData Element shall be used. If the SecurityEventContextData Defintion is structured into several elements then for each one a SecurityEventContextDataElement shall be aggregated. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
version	PositiveInteger	0..1	attr	Version number of the context data. For more details see the IDSM protocol specfication.

Table A.43: SecurityEventContextDataDefinition

Class	**SecurityEventContextDataElement**			
Note	This meta-class represents one ContextDataElement in the context of the aggregating SecurityEvent ContextDataDefintion.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventContextDataDefinition.contextDataElement, SecurityEventContextDataElement.nestedContextData			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxLength	PositiveInteger	0..1	attr	Describes the maximal length of the context data in case of Arrays and Strings.
nestedContext Data (ordered)	SecurityEventContext DataElement	*	aggr	This self-aggregation supports the desription of nested context data.
network Representation	SwDataDefProps	0..1	aggr	Definition of the networkRepresentation of the context data element. **Tags:** atp.Splitkey=networkRepresentation

▽

△

Class	**SecurityEventContextDataElement**			
securityEvent ReportInstance Definition	SecurityEventReport InstanceDefinition	0..1	ref	This reference identifies the definition of the report instance. **Tags:** atp.Splitkey=securityEventReportInstance Definition

Table A.44: SecurityEventContextDataElement

Class	***SecurityEventContextMapping*** (abstract)			
Note	This meta-class represents the ability to create an association between a collection of security events, an IdsM instance which handles the security events and the filter chains applicable to the security events.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Subclasses	SecurityEventContextMappingApplication, SecurityEventContextMappingBswModule, SecurityEvent ContextMappingCommConnector, SecurityEventContextMappingFunctionalCluster			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
filterChain	SecurityEventFilter Chain	0..1	ref	This reference defines the filter chain to be applied to each of the referenced security events (depending on the reporting mode). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=filterChain, filterChain.securityEventFilter Chain, filterChain.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
idsmInstance	IdsmInstance	0..1	ref	This reference defines the IdsmInstance onto which the security events are mapped. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=idsmInstance, idsmInstance.idsmInstance, idsmInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
mappedSecurity Event	SecurityEventContext Props	*	aggr	This aggregation represents (through further references) the SecurityEventDefinitions to be mapped to an Idsm Instance with additional mapping-dependent properties. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mappedSecurityEvent, mappedSecurity Event.shortName, mappedSecurityEvent.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.45: SecurityEventContextMapping

Class	**SecurityEventContextMappingApplication**
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context of an application (e.g. name of SWC on CP or name of SWCL on AP) in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingApplications
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*
Aggregated by	ARPackage.element

▽

△

Class	**SecurityEventContextMappingApplication**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
affected Application	String	0..1	attr	This attribute is used to identify the name of the application in whose executional context a security event can occur. This application can be, for example, a name of a Software Component (for CP) or a Software Cluster name (for AP).

Table A.46: SecurityEventContextMappingApplication

Class	**SecurityEventContextMappingFunctionalCluster**			
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context of a functional cluster in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingFunctionalClusters This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
affected Functional Cluster	String	0..1	attr	This attribute is used to identify the name of the functional cluster in whose executional context a security event can occur. The set of functional cluster names is standardized by AUTOSAR.

Table A.47: SecurityEventContextMappingFunctionalCluster

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.

▽

△

Class	**SecurityEventContextProps**			
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.48: SecurityEventContextProps

Class	**SecurityEventDefinition**			
Note	This meta-class defines a security-related event as part of the intrusion detection system. **Tags:** atp.recommendedPackage=SecurityEventDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventSymbol Name	SymbolProps	0..1	aggr	This aggregation defines optionally an alternative Event Name for the SecurityEventDefinition in case there is a collision of shortNames. **Tags:** atp.Splitkey=eventSymbolName, eventSymbol Name.shortName
id	PositiveInteger	0..1	attr	This attribute represents the numerical identification of the defined security event. The identification shall be unique within the scope of the IDS.
securityEvent ContextData Definition	SecurityEventContext DataDefinition	*	ref	Definition of additional context data that is reported with the security event in order to better support the analysis of a possible security threat. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEventContextDataDefinition, security EventContextDataDefinition.securityEventContextData Definition, securityEventContextDataDefinition.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table A.49: SecurityEventDefinition

Class	**SecurityEventFilterChain**			
Note	This meta-class represents a configurable chain of filters used to qualify security events. The different filters of this filter chain are applied in the follow order: SecurityEventStateFilter, SecurityEventOneEvery NFilter, SecurityEventAggregationFilter, SecurityEventThresholdFilter. **Tags:** atp.recommendedPackage=SecurityFilterChains			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aggregation	SecurityEvent AggregationFilter	0..1	aggr	This aggregation represents the aggregation filter in the filter chain. **Tags:** atp.Splitkey=aggregation, aggregation.shortName
oneEveryN	SecurityEventOneEvery NFilter	0..1	aggr	This aggregation represents the sampling filter in the filter chain. **Tags:** atp.Splitkey=oneEveryN, oneEveryN.shortName

▽

△

Class	**SecurityEventFilterChain**			
state	SecurityEventStateFilter	0..1	aggr	This aggregation represents the state filter in the event chain. **Tags:** atp.Splitkey=state, state.shortName
threshold	SecurityEventThresholdFilter	0..1	aggr	This aggregation represents the threshold filter in the filter chain. **Tags:** atp.Splitkey=threshold, threshold.shortName

Table A.50: SecurityEventFilterChain

Class	**SecurityEventOneEveryNFilter**			
Note	This meta-class represents the configuration of a sampling (i.e. every n-th event is sampled) filter for security events.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.oneEveryN			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
n	PositiveInteger	0..1	attr	This attribute represents the configuration of the sampling filter, i.e. it configures the parameter "n" that controls how many events (n-1) shall be dropped after a sampled event until a new sample is created.

Table A.51: SecurityEventOneEveryNFilter

Class	**SecurityEventReportInstanceDefinition**			
Note	This class shall be used to provide definition-level information for the identification of security-event context data. **Tags:** atp.recommendedPackage=SecurityEventReportInstanceDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextTypeList (ordered)	String	*	attr	This attribute shall only be used for the case that an instanceRef is required. The attribute identifies the ordered collection of the type of context elements within the instanceRef.
targetType	String	0..1	attr	This attribute shall only be used for the case that an instanceRef is required. The attribute identifies the type of the target element within the instanceRef.

Table A.52: SecurityEventReportInstanceDefinition

Class	**SecurityEventReportInstanceValue**			
Note	This class shall be used to provide information for the identification of security-event context data on the value side. **Tags:** atp.recommendedPackage=SecurityEventReportInstanceValues			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**SecurityEventReportInstanceValue**			
contextData Element Identification	SecurityEventContext DataElement	0..1	ref	This reference contributes to the identification of the context data element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=contextDataElementIdentification, context DataElementIdentification.securityEventContextData Element, contextDataElementIdentification.variation Point.shortLabel vh.latestBindingTime=systemDesignTime
flatObject	Referrable	0..1	ref	This reference shall be used if the referenced object can directly be referenced using a flat reference. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=flatObject
id	PositiveInteger	0..1	attr	This attribute represents the numerical value used for the identification of the context data element.
object	AtpFeature	0..1	iref	This reference shall be used of the target of the reference can only be identified by an instanceRef. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=object, object.contextElement, object.target **InstanceRef implemented by:** AnyInstanceRef

Table A.53: SecurityEventReportInstanceValue

Class	**SecurityEventStateFilter**			
Note	This meta-class represents the configuration of a state filter for security events. The referenced states represent a block list, i.e. the security events are dropped if the referenced state is the active state in the relevant state machine (which depends on whether the IdsM instance runs on the Classic or the Adaptive Platform).			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.state			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blockIfState ActiveAp	ModeDeclaration	*	iref	For the AP, this reference defines the machine states of the block list. That means, if a security event (mapped to the filter chain to which the SecurityEventStateFilter belongs to) is reported when the machine is in one of the block listed states, the IdsM shall discard the reported security event. **Tags:** atp.Splitkey=blockIfStateActiveAp, blockIfState ActiveAp.contextModeDeclarationGroupPrototype, blockIf StateActiveAp.targetModeDeclaration **InstanceRef implemented by:** FunctionGroupStateIn FunctionGroupSetInstanceRef This Attribute is only used by the AUTOSAR Adaptive Platform.
blockIfState ActiveCp	BlockState	*	ref	For the CP, this reference defines the states of the block list. That means, if a security event (mapped to the filter chain to which the SecurityEventStateFilter belongs to) is reported when the currently active block state in the IdsM is one of the referenced block listed states, the IdsM shall discard the reported security event. **Tags:** atp.Splitkey=blockIfStateActiveCp This Attribute is only used by the AUTOSAR Classic Platform.

Table A.54: SecurityEventStateFilter

Class	**SecurityEventThresholdFilter**			
Note	This meta-class represents the threshold filter that drops (repeatedly at each beginning of a configurable time interval) a configurable number of security events . All subsequently arriving security events (within the configured time interval) pass the filter.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.threshold			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
intervalLength	TimeValue	0..1	attr	This attribute configures the time interval in seconds for one threshold filter operation.
threshold Number	PositiveInteger	0..1	attr	This attribute configures the threshold number, i.e. how many security events in the configured time frame are dropped before subsequent events start to pass the filter.

Table A.55: SecurityEventThresholdFilter

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.56: SwBaseType

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.57: SwComponentPrototype

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. **Tags:** vh.latestBindingTime=codeGenerationTime
Base	*ARObject*

▽

△

Class	«atpVariation» **SwDataDefProps**			
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, CppImplementationDataTypeElement.swDataDefProps, *DataPrototype*.swDataDefProps, DataPrototypeTransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, DiagnosticEnvDataElementCondition.swDataDefProps, DiagnosticExtendedDataRecordElement.swDataDefProps, DiagnosticSovdPrimitiveContentElement.swDataDefProps, DltArgumentProps.networkRep, FlatInstanceDescriptor.swDataDefProps, ImplementationDataTypeElement.swDataDefProps, InstantiationDataDefProps.swDataDefProps, ISignal.networkRepresentationProps, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.swDataDefProps, *ReceiverComSpec*.networkRepresentation, SecurityEventContextDataElement.networkRepresentation, *SenderComSpec*.networkRepresentation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTargetProps.swDataDefProps, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, SystemSignal.physicalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
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.
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** atp.Splitkey=invalidValue xml.sequenceOffset=255
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** atp.Splitkey=swComparisonVariable xml.sequenceOffset=170 xml.typeElement=false

▽

△

Class	«atpVariation» **SwDataDefProps**			
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
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** atp.Splitkey=swTextProps xml.sequenceOffset=120
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

Table A.58: SwDataDefProps

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.59: SwSystemconst

Class	**System**			
Note	The top level element of the Abstract Platform System Description. **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*

▽

△

Class	**System**			
mapping	SystemMapping	*	aggr	**Stereotypes:** atpVariation **Tags:** atp.Splitkey=mapping, mapping.shortName, mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
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
systemVersion	RevisionLabelString	0..1	attr	Version number of the System Description.

Table A.60: System

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_12022]	Existence of role `IdsmRateLimitation.maxEventsInInterval`
[constr_12023]	Existence of role `IdsmRateLimitation.timeInterval`
[constr_12024]	Existence of role `SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster`
[constr_12025]	Existence of role `SecurityEventContextMappingApplication.affectedApplication`
[constr_9405]	`DltArgument`s with `variableLength` or `DltArgument.optional` settings inside of a `DltMessage`
[constr_9406]	Setting of `DltArgumentProps.variableLength` to true

Table B.1: Added Constraints in R26-11

B.1.2 Changed Constraints in R26-11

Number	Heading
[constr_5098]	Allowed `SwDataDefProps` attributes for `DltArgumentProps.networkRep`
[constr_5303]	Restriction of `baseTypeSize` of a `DltArgument`
[constr_5304]	Datatype of an Array
[constr_5305]	`CompuMethod` in `DltArgumentProps.networkRep`
[constr_5610]	Unambiguous definition of execution platform for an `IdsmInstance`
[constr_5612]	Unambiguous definition of platform-dependent network configuration for an `IdsmInstance`

Table B.2: Changed Constraints in R26-11

B.1.3 Deleted Constraints in R26-11

Number	Heading
[constr_5611]	Unambiguous configuration of platform-dependent signature support for an `IdsmInstance`

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_12004]	Valid interval for attribute `SecurityEventContextProps.sensorInstanceId`
[constr_12005]	Valid interval for attribute `IdsmInstance.idsmInstanceId`
[constr_12006]	Existence of attribute `IdsmInstance.idsmInstanceId`
[constr_12007]	Existence of attribute `IdsmInstance.rateLimitationFilter`
[constr_12008]	Existence of attribute `IdsmInstance.trafficLimitationFilter`
[constr_12009]	Existence of attribute `IdsmTrafficLimitation.maxBytesInInterval`
[constr_12010]	Existence of attribute `IdsmTrafficLimitation.timeInterval`
[constr_12011]	Existence of attribute `SecurityEventDefinition.id`
[constr_12012]	Existence of attribute `SecurityEventAggregationFilter.contextDataSource`
[constr_12013]	Existence of attribute `SecurityEventAggregationFilter.minimumIntervalLength`
[constr_12014]	Existence of attribute `SecurityEventOneEveryNFilter.n`
[constr_12015]	Existence of attribute `SecurityEventThresholdFilter.intervalLength`
[constr_12016]	Existence of attribute `SecurityEventThresholdFilter.thresholdNumber`
[constr_12017]	Existence of attribute `SecurityEventContextMapping.idsmInstance`
[constr_12018]	Existence of attribute `SecurityEventContextMapping.mappedSecurityEvent`
[constr_12019]	Existence of attribute `SecurityEventContextProps.defaultReportingMode`
[constr_12020]	Existence of attribute `SecurityEventContextProps.securityEvent`
[constr_12021]	Existence of attribute `SecurityEventContextProps.sensorInstanceId`
[constr_9364]	Mutually exclusive existence of `DltArgument.optional` and `DltArgument.predefinedText` attributes
[constr_9365]	`DltArgument` with `predefinedText`

Table B.4: Added Constraints in R25-11

B.2.2 Changed Constraints in R25-11

Number	Heading
[constr_5304]	Datatype of an Array
[constr_5305]	CompuMethod in DltArgumentProps.networkRepresentation
[constr_5363]	Allowed usage of attributes for description of payload data types
[constr_5364]	Allowed usage of attributes in case of a dltArgumentEntry
[constr_5600]	Valid interval for attribute SecurityEventDefinition.id
[constr_5602]	Valid interval for attribute SecurityEventOneEveryNFilter.n
[constr_5605]	Valid interval for attribute SecurityEventThresholdFilter.thresholdNumber
[constr_5607]	Valid interval for attribute IdsmRateLimitation.maxEventsInInterval
[constr_5609]	Valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval

Table B.5: Changed Constraints in R25-11

B.2.3 Deleted Constraints in R25-11

Number	Heading
[constr_5302]	Restriction in usage of DltArgument.optional attribute

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_12000]	Usage of references in the context of SecurityEventReportInstanceValue
[constr_12001]	Existence of attribute SecurityEventReportInstanceValue.id
[constr_12002]	Existence of attribute SecurityEventReportInstanceDefinition.targetType
[constr_12003]	Existence of reference SecurityEventReportInstanceValue.contextDataElementIdentification
[constr_9339]	SecurityEventContextDataElement.maxLength usage restriction
[constr_9340]	Datatype of an Array
[constr_9341]	CompuMethod in SecurityEventContextDataElement.networkRepresentation



△

Number	Heading
[constr_9342]	Allowed range of `SecurityEventContextDataDefinition.version`

Table B.7: Added Constraints in R24-11

B.3.2 Changed Constraints in R24-11

Number	Heading
[constr_5098]	Allowed `SwDataDefProps` attributes for `DltArgument.networkRepresentation`
[constr_5301]	Existence of `DltMessage.messageId`
[constr_5302]	Restriction in usage of `DltArgument.optional` attribute
[constr_5303]	Restriction of `baseTypeSize` of a `DltArgument`
[constr_5304]	Datatype of an Array
[constr_5305]	`CompuMethod` in `DltArgument.networkRepresentation`
[constr_5340]	Range of `DltMessage.privacyLevel.privacyLevel`
[constr_5341]	Range of `PrivacyLevel.compuMethod`
[constr_5363]	Allowed usage of attributes for description of payload data types
[constr_5364]	Allowed usage of attributes in case of a `dltArgumentEntry`

Table B.8: Changed Constraints in R24-11

B.3.3 Deleted Constraints in R24-11

none