

Document Title	Requirements on Standardization Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	536

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Document Change History			
Date	Release	Changed by	Description
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> Profiles for Data Exchange Points restructure chapters editorial changes
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> extend traceability to new document artefacts
2014-03-31	4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> editorial changes improvement of document traceability
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> editorial changes improvement of document traceability
2011-12-22	4.0.3	AUTOSAR Administration	Initial Release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1	Introduction	9
1.1	Document Conventions	9
1.2	Guidelines	10
1.3	Use Case Tracing	11
1.4	Requirements Tracing	13
2	Use Cases	15
	[UC_STDT_00001] Support Application Interfaces	15
	[UC_STDT_00002] Express Parts of SWS	15
	[UC_STDT_00003] Standardize ECUCParamdefs	16
	[UC_STDT_00004] Express predefined Paths	16
	[UC_STDT_00005] Express PlatformTypes	16
	[UC_STDT_00006] Express Examples of applied Standards	16
	[UC_STDT_00007] Support Verification if an implementation adheres to defined Standard	16
	[UC_STDT_00008] Support reusable Documentation	17
	[UC_STDT_00009] Define name conventions	17
	[UC_STDT_00010] Perform Standardization on Levels beyond the AUTOSAR Scope	17
	[UC_STDT_00011] Derive Objects from Blueprints by manually changing properties	17
	[UC_STDT_00012] Derive Objects from Blueprints in a completely standardized Way	17
	[UC_STDT_00013] Integrate compile test	17
	[UC_STDT_00014] Generate BSW "Standard AUTOSAR Interface" description from model	18
	[UC_STDT_00015] Handle General Specification Items	18
	[UC_STDT_00016] Manage requirements in AUTOSAR	18
	[UC_STDT_00017] Manage specification items in AUTOSAR	18
	[UC_STDT_00018] Manage constraint items in AUTOSAR	18
	[UC_STDT_00019] Manage test items in AUTOSAR	18
3	Requirements	19
3.1	Blueprints	19
	[RS_STDT_00001] Shall support and explain Blueprints in general	19
	[RS_STDT_00002] Formalized description of BSW SWS	19
	[RS_STDT_00003] Shall allow to represent port blueprints	20
	[RS_STDT_00004] Shall allow to represent <code>shortName</code> patterns	20
	[RS_STDT_00010] Shall refer to ECUC parameter definition	21
	[RS_STDT_00011] Shall be able to standardize components	21
	[RS_STDT_00012] Shall be able to standardize architecture	21
	[RS_STDT_00013] Shall be able to express parts of reference paths resp. package hierarchies	22
	[RS_STDT_00014] Shall be able to express levels of obligation	22

	[RS_STDT_00015] Shall support different Approaches to derive from Blueprints	22
	[RS_STDT_00017] Shall cover the compatibility of blueprints and derived objects	23
	[RS_STDT_00018] Shall allow to describe the dependencies of APIs (e.g. invocation and callback/polling interfaces)	23
	[RS_STDT_00019] Shall define the mandatory semantics for a Blueprint	24
	[RS_STDT_00020] Shall support variants of a VariableDataprototype .	24
	[RS_STDT_00021] Shall support multiple instantiation for an example SWC with PortBlueprint	24
	[RS_STDT_00022] Means of exchange format between stakeholders for blueprints	25
	[RS_STDT_00023] Shall be able to standardize Alias Names	25
	[RS_STDT_00026] Shall allow to represent port interface blueprints . .	25
	[RS_STDT_00027] Shall allow to evaluate the integrity of Blueprints . .	26
	[RS_STDT_00029] Shall be able to represent further Blueprints	26
	[RS_STDT_00030] Shall allow to standardize package structures	27
	[RS_STDT_00032] Shall be able to provide Blueprints for Roles and Rights	27
	[RS_STDT_00033] Shall be able to provide Blueprints for Build Action Manifest	28
	[RS_STDT_00034] Blueprinting of Implicit Communication Behavior . .	28
	[RS_STDT_00035] Shall support blueprinting of keywords	28
	[RS_STDT_00040] Multiplicity of elements in derived objects	28
3.2	Keywords	29
	[RS_STDT_00005] Shall support keywords and keyword abbreviations .	29
3.3	AUTOSAR Integration and Lifecycle	29
	[RS_STDT_00006] Shall be implemented without compatibility problems to existing template	29
	[RS_STDT_00007] Shall be based on the AUTOSAR XML schema	30
	[RS_STDT_00016] Shall be able to express information about the state of model elements	30
	[RS_STDT_00125] Support of AUTOSAR Specific Modeling Patterns . .	30
3.4	Traceability	31
	[RS_STDT_00008] Shall provide means to support analyzing the conformity of implementations with the AUTOSAR standards . .	31
3.5	Documentation of Specification Elements	31
	[RS_STDT_00009] Shall be able to represent requirements stated in SWS	31
	[RS_STDT_00024] Shall be able to standardize Unique Names and Display Names	32
	[RS_STDT_00025] Shall be able to standardize life cycle states	32
	[RS_STDT_00028] Shall allow to generate BSW "Standard AUTOSAR Interface" description from model	33
	[RS_STDT_00031] Shall support general specification items	33
	[RS_STDT_00036] StandardizationTemplate shall specify the representation of requirements in AUTOSAR documents	33

	[RS_STDT_00037] StandardizationTemplate shall specify the representation of specification items in AUTOSAR documents	34
	[RS_STDT_00038] StandardizationTemplate shall specify the representation of constraint items in AUTOSAR documents	34
	[RS_STDT_00039] StandardizationTemplate shall specify the representation of test items in AUTOSAR documents	34
	[RS_STDT_00041] Formalized description of BSW Abstract SWS	35
	[RS_STDT_00042] Shall provide the ability to define naming conventions for public symbols	35
3.6	Profiles for Data Exchange Points	36
	[RS_STDT_00101] Description of Data Exchange Point Shall Provide a Human Readable High-Level Overview	36
	[RS_STDT_00102] Description of Data Exchange Point Shall Describe Work Product in Methodology	36
	[RS_STDT_00103] Description of Data Exchange Point Shall Describe Intended Use	36
	[RS_STDT_00104] Description of Data Exchange Point Shall Describe Tool and Organization	37
	[RS_STDT_00105] Description of Data Exchange Point Shall Describe AUTOSAR Revision	37
	[RS_STDT_00106] Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of the AUTOSAR Meta-Model	37
	[RS_STDT_00107] Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of Model	38
	[RS_STDT_00108] Description of Data Exchange Point Shall Describe Relevant Constraints	39
	[RS_STDT_00109] Description of Data Exchange Point Shall Describe Relevant Spec Items	39
	[RS_STDT_00110] Description of Data Exchange Point Shall Describe Model Completeness	39
	[RS_STDT_00111] Description of Data Exchange Point Shall Describe Applicability of Default Values	40
	[RS_STDT_00113] Description of Data Exchange Point Shall Describe Limitation of Values of Primitive Attributes	40
	[RS_STDT_00114] Description of Data Exchange Point Shall Support Severity Levels for Compliance with Individual Rules of the Profile	41
	[RS_STDT_00115] Description of Data Exchange Point Shall Describe Rationales of Decisions	41
	[RS_STDT_00116] Description of Data Exchange Point Shall Describe Usage of AUTOSAR Extension Mechanisms	41
	[RS_STDT_00117] AUTOSAR Shall Provide Guidelines for Comparison of Profiles for Data Exchange Points	42
	[RS_STDT_00118] AUTOSAR Shall Provide Guidelines for Compatibility of Profiles for Data Exchange Points	42

[RS_STDT_00120] AUTOSAR Shall Provide Support for Handling of Incomplete Profiles for Data Exchange Points	43
[RS_STDT_00121] AUTOSAR Shall Provide Guidance for Checking Compliance of AUTOSAR Model Against Profiles for Data Exchange Points	43
[RS_STDT_00119] AUTOSAR Shall provide Rules for Composition of Profiles for Data Exchange Points	44
[RS_STDT_00122] AUTOSAR Shall Provide Guidance for Identification of Not Yet Described Aspects within Profiles for Data Exchange Points	44
[RS_STDT_00123] AUTOSAR Shall Provide Guidance for Consistency of Profiles for Data Exchange Points	44
A Change History	46
A.1 Change History R4.0.3	46
A.1.1 Added Use Cases	46
A.1.2 Added Requirements	46
A.2 Change History R4.1.1	47
A.2.1 Added Use Cases	47
A.2.2 Added Requirements	47
A.3 Change History R4.1.2	47
A.3.1 Added Use Cases	47
A.3.2 Added Requirements	47
A.4 Change History R4.1.3	48
A.4.1 Added Use Cases	48
A.4.2 Added Requirements	48
A.5 Change History R4.2.1	48
A.5.1 Added Traceables in 4.2.1	48
A.5.2 Changed Traceables in 4.2.1	48
A.5.3 Deleted Traceables in 4.2.1	49
A.6 Change History R4.2.2	49
A.6.1 Added Traceables in 4.2.2	49
A.6.2 Changed Traceables in 4.2.2	50
A.6.3 Deleted Traceables in 4.2.2	50
A.7 Change History R4.3.0	50
A.7.1 Added Traceables in 4.3.0	50
A.7.2 Changed Traceables in 4.3.0	51
A.7.3 Deleted Traceables in 4.3.0	51
A.8 Change History R4.3.1	51
A.8.1 Added Traceables in 4.3.1	51
A.8.2 Changed Traceables in 4.3.1	51
A.8.3 Deleted Traceables in 4.3.1	51
B Mentioned Class Tables	52

References

- [1] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate
- [2] Standardization Template
AUTOSAR_TPS_StandardizationTemplate
- [3] Requirements on AUTOSAR Features
AUTOSAR_RS_Features
- [4] Main Requirements
AUTOSAR_RS_Main
- [5] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration
- [6] Specification of Platform Types
AUTOSAR_SWS_PlatformTypes
- [7] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate
- [8] XML Specification of Application Interfaces
AUTOSAR_MOD_AISpecification
- [9] SW-C and System Modeling Guide
AUTOSAR_TR_SWCModelingGuide
- [10] Requirements on Interoperability of AUTOSAR Tools
AUTOSAR_RS_InteroperabilityOfAutosarTools

1 Introduction

AUTOSAR models are in many cases not created from scratch but existing content is taken as the basis. The existing content could be contributed by the AUTOSAR initiative itself in form of standardized model elements.

This document specifies the requirements for the Standardization Template. This template is intended to support the delivery of standardized model elements by AUTOSAR.

AUTOSAR 4.0 already specifies the blueprint approach for standardization. This approach is continued and refined by the Standardization Template. It thereby will replace Appendix A in Software Component Template [1].

As an particular example, let us consider the standardization of application interfaces. That is, in terms of the AUTOSAR meta-model the standardization mainly applies to the definition of `PortPrototypes` for specific purposes.

Due to the structure of the AUTOSAR meta-model it is not possible to merely express a standardized `PortPrototype` because for good reasons the latter does not exist on its own but is always owned by a `SwComponentType`.

The Standardization Template specifies the approach to overcome this situation.

The referenced deliverable AUTOSAR_RS_InteroperabilityOfAutosarTools is set to status obsolete in release 4.4.0

1.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([2]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([2]).

1.2 Guidelines

Existing specifications shall be referenced (in form of a single requirement). Differences to these specifications are specified as additional requirements. All Requirements shall have the following properties:

- **Redundancy**
Requirements shall not be repeated within one requirement or in other requirements.
- **Clearness**
All requirements shall allow one possibility of interpretation only. Used technical terms that are not in the glossary must be defined.
- **Atomicity**
Each Requirement shall only contain one requirement. A Requirement is atomic if it cannot be split up in further requirements.
- **Testability**
Requirements shall be testable by analysis, review or test.
- **Traceability**
The source and status of a requirement shall be visible at all times.

1.3 Use Case Tracing

Following table references the use cases specified and links to the related requirements.

Use Case	Description	Satisfied by
[UC_STDT_00001]	Support of Application Interfaces	[RS_STDT_00001] [RS_STDT_00003] [RS_STDT_00005] [RS_STDT_00006] [RS_STDT_00007] [RS_STDT_00016] [RS_STDT_00019] [RS_STDT_00020] [RS_STDT_00021] [RS_STDT_00022] [RS_STDT_00026] [RS_STDT_00035]
[UC_STDT_00002]	Express parts of SWS	[RS_STDT_00001] [RS_STDT_00002] [RS_STDT_00018] [RS_STDT_00041]
[UC_STDT_00003]	Standardize ECUC Parameters	[RS_STDT_00001] [RS_STDT_00010] [RS_STDT_00029]
[UC_STDT_00004]	Express predefined Paths	[RS_STDT_00001] [RS_STDT_00013] [RS_STDT_00030]
[UC_STDT_00005]	Express Platform Types	[RS_STDT_00001]
[UC_STDT_00006]	Express Examples of applied Standards	[RS_STDT_00001]
[UC_STDT_00007]	Support Verification if an Implementation adheres to defined Standard	[RS_STDT_00001] [RS_STDT_00008] [RS_STDT_00009] [RS_STDT_00015] [RS_STDT_00017]
[UC_STDT_00008]	Support reusable Documentation	[RS_STDT_00001] [RS_STDT_00002] [RS_STDT_00003] [RS_STDT_00023] [RS_STDT_00026] [RS_STDT_00041]
[UC_STDT_00009]	Define name Conventions	[RS_STDT_00001] [RS_STDT_00004] [RS_STDT_00014] [RS_STDT_00023] [RS_STDT_00024] [RS_STDT_00025] [RS_STDT_00042]
[UC_STDT_00010]	STDT shall be applicable beyond the Scope of AUTOSAR	[RS_STDT_00011] [RS_STDT_00012] [RS_STDT_00024] [RS_STDT_00025] [RS_STDT_00032] [RS_STDT_00033]

Use Case	Description	Satisfied by
[UC_STDT_00011]	Derive objects from Blueprints by adding missing Information	[RS_STDT_00015] [RS_STDT_00029] [RS_STDT_00040]
[UC_STDT_00012]	Derive Objects from Blueprints in a completely standardized Way	[RS_STDT_00015] [RS_STDT_00029] [RS_STDT_00040]
[UC_STDT_00013]	Integrate compile test	[RS_STDT_00027]
[UC_STDT_00014]	Generate BSW "Standard AUTOSAR Interface" description from model	[RS_STDT_00028]
[UC_STDT_00015]	Handle General Specification Items	[RS_STDT_00031] [RS_STDT_00041]
[UC_STDT_00016]	Manage requirements in AUTOSAR	[RS_STDT_00036]
[UC_STDT_00017]	Manage specification items in AUTOSAR	[RS_STDT_00037]
[UC_STDT_00018]	Manage constraint items in AUTOSAR	[RS_STDT_00038]
[UC_STDT_00019]	Manage test items in AUTOSAR	[RS_STDT_00039]

Table 1.1: Use Case Tracing

1.4 Requirements Tracing

The following table references the requirements specified in [3], [4] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_BRF_01024]	AUTOSAR shall provide naming rules for public symbols	[RS_STDT_00042]
[RS_BRF_01056]	AUTOSAR BSW modules shall provide standardized interfaces	[RS_STDT_00028]
[RS_BRF_01064]	AUTOSAR BSW shall provide callback functions in order to access upper layer modules	[RS_STDT_00018]
[RS_BRF_04000]	AUTOSAR documentation shall support traceability	[RS_STDT_00013] [RS_STDT_00030] [RS_STDT_00041]
[RS_BRF_04008]	AUTOSAR documentation shall support consistency and quality assurance	[RS_STDT_00040]
[RS_BRF_04016]	AUTOSAR shall support modeling and documentation guidelines	[RS_STDT_00002] [RS_STDT_00004] [RS_STDT_00005] [RS_STDT_00006] [RS_STDT_00007] [RS_STDT_00008] [RS_STDT_00009] [RS_STDT_00011] [RS_STDT_00012] [RS_STDT_00014] [RS_STDT_00016] [RS_STDT_00020] [RS_STDT_00023] [RS_STDT_00024] [RS_STDT_00025] [RS_STDT_00031] [RS_STDT_00036] [RS_STDT_00037] [RS_STDT_00038] [RS_STDT_00039]
[RS_BRF_04024]	AUTOSAR shall support guidance for applying the specifications	[RS_STDT_00001] [RS_STDT_00003] [RS_STDT_00010] [RS_STDT_00015] [RS_STDT_00017] [RS_STDT_00019] [RS_STDT_00021] [RS_STDT_00022] [RS_STDT_00026] [RS_STDT_00027] [RS_STDT_00029] [RS_STDT_00032] [RS_STDT_00033] [RS_STDT_00034] [RS_STDT_00035]

Requirement	Description	Satisfied by
[RS_Main_00300]	AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups	[RS_STDT_00101] [RS_STDT_00102] [RS_STDT_00103] [RS_STDT_00104] [RS_STDT_00105] [RS_STDT_00106] [RS_STDT_00107] [RS_STDT_00108] [RS_STDT_00109] [RS_STDT_00110] [RS_STDT_00111] [RS_STDT_00113] [RS_STDT_00114] [RS_STDT_00115] [RS_STDT_00116] [RS_STDT_00117] [RS_STDT_00118] [RS_STDT_00119] [RS_STDT_00120] [RS_STDT_00121] [RS_STDT_00122] [RS_STDT_00123] [RS_STDT_00125]

Table 1.2: Requirements Tracing

2 Use Cases

This chapter describes use-cases for the Standardization Template. The intention of these uses cases is to point out the potential applications of the Standardization Template. In general, the use case of the Standardization Template is to express items standardized by AUTOSAR as AUTOSAR XML artifact. This artifact can subsequently be used to support the development of AUTOSAR compliant products.

Each use-case defined in this document has its unique identifier starting with the prefix "UC_STDT_" (meaning Standardization Template Use Case).

[UC_STDT_00001] Support Application Interfaces [AUTOSAR : provides standardized, openly disclosed interfaces for different domains such as chassis, powertrain, body etc. The definition of these interfaces can be handled on various levels of deepness:

- L8** Complete description of the SW-Cs including behavioral model and ports
 - L7** Complete description of all ports including interface behavior and data qualities
 - L6** Complete description of all ports including textual description of interface behavior and the data qualities of the interfaces
 - L5** Complete description of all ports including data qualities of the interfaces
 - L4** Complete description of all ports including within-AUTOSAR-agreed data qualities
 - L3** Partial description of ports/interfaces of a SWC including within-AUTOSAR-agreed data qualities
- Note that this partial description includes the fact that only some of the ports are described, as well as the fact that this description of a port is incomplete and is also separated from the applicable component. This is also known as PortBlueprint.
- L2** Dictionary of interfaces including a set of within-AUTOSAR-agreed data qualities
 - L1** Dictionary of data elements including types and ranges.
 - L0** Dictionary of names

As of Release 4.0, AUTOSAR standardization covers the level L0 ... L3. Nevertheless vendor internal applications might use Standardization Template for higher levels too.

This use case mainly covers the application software aspects.

Applying this formal description will improve consistency and usability of the AUTOSAR Application Interfaces and empower formal checks e.g. of backward compatibility of application interfaces.]()

[UC_STDT_00002] Express Parts of SWS [The Standardization Template shall allow to express parts of SWS for basic software modules formally using the AUTOSAR schema. This includes (but is not restricted to):

- Standardized interfaces (i.e. C-APIs)
- Standardized AUTOSAR Interfaces (Ports, PortInterfaces, ...)
- Definition of ECUC-Parameters (see [UC_STDT_00003])

Applying this formal description will improve consistency and usability of the AUTOSAR SWS and empower formal checks e.g. of backward compatibility of interfaces.]()

[UC_STDT_00003] Standardize ECUCParamdefs [Part of the AUTOSAR SWS is also the set of ECU configuration parameter definitions. These parameter definitions are the basis of the so called vendor specific parameters which are used in particular AUTOSAR implementations.

Even if this is specified in great detail in [5] it is also in the scope of Standardization Template.]()

[UC_STDT_00004] Express predefined Paths [Development partners may mutually agree on a particular package layout and thus share AUTOSAR artifacts in a later phase of the development. For this use case it is helpful to initially express a set of predefined resp. partly predefined reference paths respective `referenceBase` which can be loaded in individual AUTOSAR authoring tools.

This use case covers the beginning or the end of a reference path. For example the usecase is to standardize the substructure after a variable path: `<My-Path>/PortInterfaces`. In this case only `PortInterfaces` is standardized.]()

[UC_STDT_00005] Express PlatformTypes [The platform types defined in [6] need to be available in processable format for AUTOSAR development tools. This approach improves consistency and quality of AUTOSAR products.

The details of [7] chapter 3.1 apply.]()

[UC_STDT_00006] Express Examples of applied Standards [In addition to the application interfaces [8] AUTOSAR provides examples how to apply the standardized elements, in particular blueprints.

The details of [7] chapter 3.1 apply.]()

[UC_STDT_00007] Support Verification if an implementation adheres to defined Standard [When an AUTOSAR product is developed an initial verification can be performed by verifying the product against the formalized standard. This includes

- the check of compatibility rules between blueprints and derived model elements. These compatibility rules are to be defined for each meta class eligible to blueprints.
- tracing between model elements and SWS respectively blueprints in order to check if all blueprints were implemented.

Note that this use case is a very initial verification and does not compete or even replace conformance test specification. It rather contributes to conformance test.

The compatibility rules need only to be described in the document, e.g. in form of constraints. There is no formal representation of the compatibility rules in the meta-model.

The compatibility rules are specified individually for each meta-class eligible for blueprinting. For example all port blueprints follow the same compatibility rules.]()
()

[UC_STDT_00008] Support reusable Documentation [Parts of the AUTOSAR SWS may be published such that it can be reused for the actual product documentation. The vendor of an implementation then takes such parts out of an Instance of a Standardization Template and incorporates it in his own software documentation.

The same approach may apply to the explanation of Application Interfaces.]()
()

[UC_STDT_00009] Define name conventions [AUTOSAR has modeling guides and naming conventions. If these conventions are published as instance of the Standardization Template, they can be utilized to configure e.g. modeling tools.

The use case also covers the ability to express various levels of obligation. This may for example be expressed similar to the keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL".]()
()

[UC_STDT_00010] Perform Standardization on Levels beyond the AUTOSAR Scope [The Standardization Template shall be applicable for company internal standardization respectively for mutual agreements which go beyond the AUTOSAR standardization on meta level¹ M1 (see [UC_STDT_00001]).]()
()

[UC_STDT_00011] Derive Objects from Blueprints by manually changing properties [The user makes a kind of copy of the blueprint and is allowed to add his own things (e.g. adding own field to a standardized enum-type). Example is the usage of ApplicationInterfaces and ConfigurationParameters.

In case of Standardized PortInterfaces of AUTOSAR Services the Standardization template shall mention the use case that in this case a PortInterface based on the "Standardized PortInterface Blueprint" might contain a subset of ClientServerOperations.]()
()

[UC_STDT_00012] Derive Objects from Blueprints in a completely standardized Way [The user can only configure or otherwise influence the content of the copied "blueprint" in a completely standardized way, (e.g. configuring the fields of an enum-type according to the "needs" of the software) but he cannot add own things.

This could even go so far, that only the rules of configuration are standardized (like in the case of DCM-PortInterfaces) - but this nonetheless completely determines the outcome in a concrete project.]()
()

[UC_STDT_00013] Integrate compile test [Until Release 4.0 all APIs of the BSW are modeled and chapter 8 of the SWS is mainly generated out of the model. Additionally

¹For more details of Meta levels see Chapter 2.2 in [7]

we propose to generate empty C functions (and data structures/consts/...) out of the model and link all these functions together. If the compile or link process fails the consistency (e.g. between different SWS) is violated and needs to be fixed.]()

[UC_STDT_00014] Generate BSW "Standard AUTOSAR Interface" description from model [Until Release 4.0 the "Standard AUTOSAR Interface" is part of each SWS which offers this interface (Typically contained in an own chapter of sub chapter of 7 or 8). The description is mostly plain text with some pseudo language to show the usage of the interface (including constants, etc.). Furthermore the description of the services often uses "elements" from the meta-model which are not up-to-date or their meaning has changed.

It is intended to standardize this part of the SWS, e.g. via an own model (and then the generated descriptions can be imported into the SWS like chapter 8) OR via a -standardized- language to clarify the understanding of the interface and allow an automatic conversation for RTE purposes.]()

[UC_STDT_00015] Handle General Specification Items [There might be a set of general requirements which need to be fulfilled by all SWS documents. On the other hand there might be a general specification which summarizes all common specification items. This situation shall be handled by tracing:

- Tracing shall use both requirements documents, the general one as well as the individual one
- General Requirements may be satisfied by the general specification or by the individual specifications.
- General Requirements may not be applicable for a particular specification.
- General Requirements may be fully satisfied only by both the general together with an individual specification.

]()

[UC_STDT_00016] Manage requirements in AUTOSAR [In AUTOSAR all requirements are formally captured in requirement documents (RS/Feature/SRS) with a unique id. Specification documents (SWS) contain specification items that formally trace to requirements. Dependencies between requirements on the same level are expressed in the requirement block itself by providing references to the requirements.]()

[UC_STDT_00017] Manage specification items in AUTOSAR [In AUTOSAR all specification items are formally captured in specification documents (TPS, SWS) with a unique id. Specification items formally trace to requirements.]()

[UC_STDT_00018] Manage constraint items in AUTOSAR [In AUTOSAR all constraint items are formally captured in specification documents (TPS, SWS) with a unique id.]()

[UC_STDT_00019] Manage test items in AUTOSAR [In AUTOSAR all test items are formally captured in specification documents with a unique id.]()

3 Requirements

This chapter describes all requirements driving the work to define the *Standardization Template* specification [2].

Each requirement in this document has its unique identifier starting with the prefix "RS_STDT_" (meaning Requirement Specification for Standardization Template).

3.1 Blueprints

[RS_STDT_00001] Shall support and explain Blueprints in general [

Type:	valid
Description:	<p>The standardization template shall support blueprints. Blueprint is a kind of "incomplete" model" which is copied and refined lateron. The principles of blueprints shall be defined:</p> <ul style="list-style-type: none"> • "Instantiation" is done by copy rather than referenced. Downstream processing excludes the usage of blueprints. • Define proper terminology for blueprints and blueprinted model elements. • How are the elements named that are created out of blueprints? • Shall clearly define which parts of the meta-model are eligible for blueprinting. Blueprinting non-eligible parts of the meta-model shall count as a "validation error". • define the rules how to derive objects from blueprints, in particular the strategy, which properties may be added / removed / redefined. These rules shall be defined individually for each meta-class eligible for blueprinting. <p>Necessary facilities of the Meta Model shall be defined:</p> <ul style="list-style-type: none"> • specific blueprints • mapping blueprints and derived objects
Rationale:	This helps to understand the concept and application of blueprints as blueprints are the main mean of standardization.
Use Case:	[UC_STDT_00001],[UC_STDT_00002],[UC_STDT_00003],[UC_STDT_00004] [UC_STDT_00005],[UC_STDT_00006],[UC_STDT_00007],[UC_STDT_00008] [UC_STDT_00009]
Dependencies:	—
Supporting Material:	—

](RS_BRF_04024)

[RS_STDT_00002] Formalized description of BSW SWS [

Type:	valid
Description:	<p>The standardization Template shall be able to publish formalized parts of a SWS which then acts as a blueprint of the specified Module.</p> <p>The Standardization Template shall provide means to describe standardized Interfaces (C-APIs).</p> <p>The Standardization Template shall allow the description standardized AUTOSAR Interfaces.</p> <p>The Standardization Template must support the specification of variants of the interfaces.</p>
Rationale:	<p>Especially the "Standard AUTOSAR Interface" is part of each SWS which offers this interface (Typically contained in an own chapter of subchapter of 7 or 8). The description is mostly plain text with some pseudo language to show the usage of the interface (including constants, etc.). Furthermore the description of the services often uses "elements" from the meta-model which are not up-to-date or their meaning has changed. The current state causes several problems when the RTE "routes" calls between the BSWs and SWC. The pseudo language must be manually transferred into some sort of "SWC-Description". If people try to mix modules from different vendors it is not clear how this can work. In our understanding the format needs to be standardized. We propose to standardize this part of the SWS, e.g. via an own model (and then the generated descriptions can be imported into the SWS like chapter 8) OR via a -standardized- language to clarify the understanding of the interface and allow an automatic conversation for RTE purposes.</p>
Use Case:	[UC_STDT_00002],[UC_STDT_00008]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00003] Shall allow to represent port blueprints [

Type:	valid
Description:	AUTOSAR standardizes so called "Application Interfaces". These Interfaces in fact result in port blueprints.
Rationale:	AUTOSAR publishes standardized Models as ARXML.
Use Case:	[UC_STDT_00001],[UC_STDT_00008]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00004] Shall allow to represent shortName patterns [

Type:	valid
Description:	Blueprints shall allow to represent <code>shortName</code> patterns that describes how to determine the <code>shortName</code> of an derived element.
Rationale:	AUTOSAR publishes the Application Interfaces Modeling guide
Use Case:	[UC_STDT_00009]
Dependencies:	TR_SWCModelingGuide [9] might need to be adapted.
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00010] Shall refer to ECUC parameter definition [

Type:	valid
Description:	<p>ECUC parameter definitions are also standardized by AUTOSAR. Therefore it is in the scope of the Standardization Template. Strictly speaking, these standardizes ECUC Parameter Definitions act as blueprints for the vendor specific parameters, even if these are not mapped using the <code>BluePrintMappingSet</code>.</p> <p>Standardization Template shall not change the approaches at least for AUTOSAR 4.0, but reflect the relationships.</p>
Rationale:	This maintains the overall scope and the applied patterns.
Use Case:	[UC_STDT_00003]
Dependencies:	–
Supporting Material:	[5]

](RS_BRF_04024)

[RS_STDT_00011] Shall be able to standardize components [

Type:	valid
Description:	<p>STDT shall be able to express standardization of components, even if AUTOSAR does not standardize components.</p> <p>This requirement covers a set of individual components.</p> <p>No compatibility rules shall hardwired in STDT. Support is provided by the fact that it only allows to specify <code>SwComponentType</code> as eligible for blueprinting.</p>
Rationale:	This allows to leverage AUTOSAR standardization principles inside a company.
Use Case:	[UC_STDT_00010]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00012] Shall be able to standardize architecture [

Type:	valid
Description:	The Standardization Template shall be able to express standardization of components and communication, even if AUTOSAR does not standardize architecture of application software. This requirement covers a set of communicating components.
Rationale:	This allows to leverage AUTOSAR standardization principles inside a company.
Use Case:	[UC_STDT_00010]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00013] Shall be able to express parts of reference paths resp. package hierarchies [

Type:	valid
Description:	The Standardization Template shall be able to express standardized reference paths resp. parts of reference paths. It shall be possible to specify the beginning respectively the end of a package hierarchy. See UC_STDT_004 for an example.
Rationale:	This allows to leverage AUTOSAR standardization principles inside a company.
Use Case:	[UC_STDT_00004]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04000)

[RS_STDT_00014] Shall be able to express levels of obligation [

Type:	valid
Description:	The level of obligation shall be expressed by key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL".
Rationale:	This allows to use Standardization Models to evaluate conformity of an implementation
Use Case:	[UC_STDT_00009]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00015] Shall support different Approaches to derive from Blueprints [

Type:	valid
Description:	<p>Shall support different approaches to derive from blueprints.</p> <p>Such different approaches are</p> <ul style="list-style-type: none"> 0. The user makes a kind of copy of the blueprint and is allowed to add his own things (e.g. adding own field to a standardized enum-type). 0. The user can only configure or otherwise influence the content of the copied "blueprint" in a completely standardized way, (e.g. configuring the fields of an enum-type according to the "needs" of the software) but he cannot add own things.
Rationale:	This allows to use Standardization Models to evaluate conformity of an implementation. Conformity depends on the approach to derive objects
Use Case:	[UC_STDT_00007],[UC_STDT_00011],[UC_STDT_00012]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00017] Shall cover the compatibility of blueprints and derived objects

Type:	valid
Description:	The Standardization Template shall describe the compatibility rules between blueprints and derived objects. These compatibility shall be described individually for each meta class eligible for blueprinting.
Rationale:	This supports a continues evolution of a standard
Use Case:	[UC_STDT_00007]
Dependencies:	[RS_STDT_00001]
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00018] Shall allow to describe the dependencies of APIs (e.g. invocation and callback/polling interfaces)

Type:	valid
Description:	The Standardization Template shall allow to describe the dependencies of invocation interfaces and the corresponding callback or polling interfaces.
Rationale:	Standardized interfaces consist in many cases of invocation interfaces (C-APIs) and callback or polling interfaces. In many cases it is configurable, which communication pattern is used. This configurable dependency and the parameters shall be described via blueprints.



△

Use Case:	[UC_STDT_00002]
Dependencies:	[RS_STDT_00002]
Supporting Material:	–

](RS_BRF_01064)

[RS_STDT_00019] Shall define the mandatory semantics for a Blueprint [

Type:	valid
Description:	<p>For a given model element, the template must define which attributes of the model element must be standardized to be entitled as a blueprint.</p> <p>For e.g which information of a PortInterface must be present to be called as blueprint ?</p> <p>In case of Standardized PortInterfaces of AUTOSAR Services the Standardization template shall mention the use case that in this case a PortInterface based on the "Standardized PortInterface Blueprint" might contain a subset of ClientServeOperations.</p>
Rationale:	Helps to have a common understanding on blueprint model element.
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00020] Shall support variants of a VariableDataprototype [

Type:	valid
Description:	The PortBlueprint should be able to map to different VariableDataprototype of the same instance of PortInterface.
Rationale:	Variant handling for WP10.3 is mostly intended for reusing the definition between passenger cars and trucks. Therefore it's probably useful to have variants at data type level instead of creating new blueprints.
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00021] Shall support multiple instantiation for an example SWC with PortBlueprint [

Type:	valid
Description:	It should be possible for the PortBlueprint to support multiple instantiation.
Rationale:	–
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00022] Means of exchange format between stakeholders for blueprints [

Type:	valid
Description:	AUTOSAR methodology shall define the exchange of the PortInterfaceMapping for a given SWCdescription file, i.e The RTE and the VFB in principle ignores the blueprints but how should the exchange be established between the stakeholders with the blueprint information, while creating the PortPrototypes out of it?.
Rationale:	–
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00023] Shall be able to standardize Alias Names [

Type:	valid
Description:	STDT shall be able to standardize alias names.
Rationale:	e.g. used for system constants in measurement and calibration system Necessary if system constants will be standardized in future (what is not yet decided)
Use Case:	[UC_STDT_00008],[UC_STDT_00009]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00026] Shall allow to represent port interface blueprints [

Type:	valid
Description:	AUTOSAR standardizes so called "Application Interfaces". These Interfaces in fact result in port blueprints and appropriate port interface blueprints
Rationale:	AUTOSAR publishes standardized Models as ARXML.
Use Case:	[UC_STDT_00001],[UC_STDT_00008]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00027] Shall allow to evaluate the integrity of Blueprints [

Type:	valid
Description:	Until Release 4.0 all APIs of the BSW are modeled and chapter 8 of the SWS is mainly generated out of the model. Additionally we propose to generate empty C functions (and data structures/consts/...) out of the model and link all these functions together. If the compile or link process fails the consistency (e.g. between different SWS) is violated and needs to be fixed.
Rationale:	In the past we had often problems that some SWS assume specific services (of structs/consts/...) from other modules and the interface did not match (or even did not exists at all). In the compile test such errors will be found.
Use Case:	[UC_STDT_00013]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04024)

[RS_STDT_00029] Shall be able to represent further Blueprints [

Type:	valid
Description:	<p>STDT shall be able to represent blueprints for the following elements:</p> <ul style="list-style-type: none"> • AliasNameSet (see RS_STDT_00023) • ApplicationDatatype • BswModuleEntry • BaseType • BswModuleDescription • CompuMethod - to enhance enumerators • DataConstr - to widen ranges as long as it fits in the BaseType not allowed to restrict : Wp: Not allowed to change the range



△

	<ul style="list-style-type: none"> • DatatypeMappingSet - mapped types in derived Mappings set must be the derived ones from blueprint • EcucModuleDef • EcucDefintionCollection • ImplementationDatatype • ModeDeclarationGroup - to add additional modes • PortInterfaces (for sender receiver and client server interfaces) (see RS_STDT_00026) • PortPrototypeBlueprints (see RS_STDT_00003) • SwComponentType
Rationale:	—
Use Case:	[UC_STDT_00003],[UC_STDT_00011],[UC_STDT_00012]
Dependencies:	See also [RS_STDT_00003], [RS_STDT_00026]
Supporting Material:	—

]([RS_BRF_04024](#))

[RS_STDT_00030] Shall allow to standardize package structures [

Type:	valid
Description:	STDT shall be able to represent blueprints of package structures in particular to predefine access paths.
Rationale:	—
Use Case:	[UC_STDT_00004]
Dependencies:	—
Supporting Material:	—

]([RS_BRF_04000](#))

[RS_STDT_00032] Shall be able to provide Blueprints for Roles and Rights [

Type:	valid
Description:	Standardization Template shall support blueprinting of roles and rights
Rationale:	—
Use Case:	[UC_STDT_00010]
Dependencies:	—
Supporting Material:	—

]([RS_BRF_04024](#))

[RS_STDT_00033] Shall be able to provide Blueprints for Build Action Manifest [

Type:	valid
Description:	Standardization Template shall support blueprinting of Processor Manifest.
Rationale:	–
Use Case:	[UC_STDT_00010]
Dependencies:	–
Supporting Material:	–

]([RS_BRF_04024](#))

[RS_STDT_00034] Blueprinting of Implicit Communication Behavior [

Type:	valid
Description:	The AUTOSAR Templates and Methodology shall support blueprinting of the Implicit Communication Behavior descriptions. Grouping of data shall be possible before the RunnableEntities with all the details (data access points) are known. In a top down approach the grouping of DataPrototypes can already be used to design the system in a way that consistency properties are guaranteed and that consistency is not required for unrelated DataPrototypes .
Rationale:	Define Implicit Communication Behavior requirements in a top down design approach
Use Case:	–
Dependencies:	–
Supporting Material:	–

]([RS_BRF_04024](#))

[RS_STDT_00035] Shall support blueprinting of keywords [

Type:	valid
Description:	Keywords shall be blueprintable in order to support vendor specific extensions.
Rationale:	Support AUTOSAR publication
Use Case:	[UC_STDT_00001]
Dependencies:	TR_SWCMModelingGuide [9] might need to be adapted.
Supporting Material:	–

]([RS_BRF_04024](#))

[RS_STDT_00040] Multiplicity of elements in derived objects [

Type:	valid
Description:	The standardization template shall support for elements with upper multiplicity > 1 the description of the expected number of derived objects.
Rationale:	This supports the task Derive From Blueprint.
Use Case:	[UC_STDT_00011],[UC_STDT_00012]
Dependencies:	—
Supporting Material:	—

](RS_BRF_04008)

3.2 Keywords

[RS_STDT_00005] Shall support keywords and keyword abbreviations [

Type:	valid
Description:	In [9] AUTOSAR publishes building rules for <code>ShortName</code> as sequence of Keywords. These keywords need to be expressed by Standardization Template. The existing Keyword identifiable should be extended with a <code>ShortLabel</code> . Semantics of <code>SHORT-LABEL</code> : used instead of <code>ShortName</code> . Necessary in case there are several name parts that are to be abbreviated by the same keyword abbreviation. As Keywords are identifiable, this would lead to a conflict.
Rationale:	Support AUTOSAR publication
Use Case:	[UC_STDT_00001]
Dependencies:	TR_SWCModelingGuide [9] might need to be adapted.
Supporting Material:	—

](RS_BRF_04016)

3.3 AUTOSAR Integration and Lifecycle

[RS_STDT_00006] Shall be implemented without compatibility problems to existing template [

Type:	valid
Description:	New templates shall ensure that they are compatible with existing templates.
Rationale:	Maintenance of backwards compatibility of the Schema as requested for 4.0
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00007] Shall be based on the AUTOSAR XML schema [

Type:	valid
Description:	The Standardization Template shall be described using the same mechanisms as the other templates. This includes the modelling in the AUTOSAR meta model and the specification of the XML representation in the AUTOSAR XML Schema.
Rationale:	General approach of having one Meta Model for all templates
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00016] Shall be able to express information about the state of model elements [

Type:	valid
Description:	It would be beneficial, if the STDT would also support information about the state of model elements. Example: Due to backward compatibility it will be difficult to delete exiting application interfaces for future releases. If some of the model elements are no longer "state of the art", they can be marked as being e.g. "obsolete".
Rationale:	This supports a continues evolution of a standard
Use Case:	[UC_STDT_00001]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00125] Support of AUTOSAR Specific Modeling Patterns [

Type:	valid
Description:	<p>The template for describing profiles for data exchange points shall be able to handle AUTOSAR specific modeling patterns such as</p> <ul style="list-style-type: none"> • VariationPoints • Splitable • Pseudo Primitive Datatypes (e.g. Identifier, Verbatim String, Limit) • Type / InstanceRef references • Relative references in XML • xml.sequenceOffset • mixed and mixedString • Formulas
Rationale:	–
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

3.4 Traceability

[RS_STDT_00008] Shall provide means to support analyzing the conformity of implementations with the AUTOSAR standards [

Type:	valid
Description:	The Standardization Template should enable the implementer to check if its description (e.g. BSWM) is in conformance with AUTOSAR standards specified on M1 level (SWS).
Rationale:	This establishes traceability between AUTOSAR Implementations and defined standard. And is also a precondition to check the application compatibility between different releases.
Use Case:	[UC_STDT_00007]
Dependencies:	–
Supporting Material:	–

]([RS_BRF_04016](#))

3.5 Documentation of Specification Elements

[RS_STDT_00009] Shall be able to represent requirements stated in SWS [

Type:	valid
Description:	To improve requirements traceability a formalized description of a SWS shall contain textual representatives of each requirement contained in the respective SWS document (specification items of SWSs). The statements shall be categorized as one of {Requirement, Specification, Implementation, Constraint}
Rationale:	This feature enables a automated tracking of changes of requirements, which is the basis for a improved change management and compatibility assessment.
Use Case:	[UC_STDT_00007]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00024] Shall be able to standardize Unique Names and Display Names [

Type:	valid
Description:	STDT shall be able to standardize Unique Names and Display Names e.g. in documentation (the complete instance reference would not be readable) and in measurement and calibration systems (standardized measurement and calibration formats like A2L require unique names for sw signals).
Rationale:	support standardization of unique names, e.g. for <ul style="list-style-type: none"> • documentation • calibration and measurement tools
Use Case:	[UC_STDT_00009],[UC_STDT_00010]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00025] Shall be able to standardize life cycle states [

Type:	valid
Description:	STDT shall be able to standardize life the states of a particular lifecycle.
Rationale:	Since AUTOSAR has the goal of being backward compatible it is not possible to just delete a standardized model element and add a new one or - even worse - to change the model element without notification.
Use Case:	[UC_STDT_00009],[UC_STDT_00010]
Dependencies:	–



△

Supporting Material:	–
-----------------------------	---

](RS_BRF_04016)

[RS_STDT_00028] Shall allow to generate BSW "Standard AUTOSAR Interface" description from model [

Type:	valid
Description:	<p>"Standard AUTOSAR Interface" is part of each SWS which offers this interface (Typically contained in an own chapter of subchapter of 7 or 8). The description is mostly plain text with some pseudo language to show the usage of the interface (including constants, etc.). Furthermore the description of the services often uses "elements" from the meta-model which are not up-to-date or their meaning has changed.</p> <p>STDT shall provide support to standardize this part of the SWS, e.g. via an own model (and then the generated descriptions can be imported into the SWS like chapter 8) OR via a -standardized- language to clarify the understanding of the interface and allow an automatic conversation for RTE purposes.</p>
Rationale:	–
Use Case:	[UC_STDT_00014]
Dependencies:	–
Supporting Material:	–

](RS_BRF_01056)

[RS_STDT_00031] Shall support general specification items [

Type:	valid
Description:	support the explicit indication of non applicable requirements and complementary specification items. In particular allow the specific trace-Indexes "NA" and "SPEC".
Rationale:	–
Use Case:	[UC_STDT_00002],[UC_STDT_00015]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00036] StandardizationTemplate shall specify the representation of requirements in AUTOSAR documents [

Type:	valid
Description:	Specify content and preferred graphical representation of structured requirements in AUTOSAR documents and AUTOSAR meta model.
Rationale:	Consistent specification and representation of requirements and specification items in AUTOSAR.
Use Case:	[UC_STDT_00016]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00037] StandardizationTemplate shall specify the representation of specification items in AUTOSAR documents [

Type:	valid
Description:	Specify content of specification items in AUTOSAR documents and AUTOSAR meta model.
Rationale:	Consistent specification and representation of specification items in AUTOSAR.
Use Case:	[UC_STDT_00017]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00038] StandardizationTemplate shall specify the representation of constraint items in AUTOSAR documents [

Type:	valid
Description:	Specify content of constraint items in AUTOSAR documents and AUTOSAR meta model.
Rationale:	Consistent specification and representation of constraint items in AUTOSAR.
Use Case:	[UC_STDT_00018]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00039] StandardizationTemplate shall specify the representation of test items in AUTOSAR documents [

Type:	valid
Description:	Specify content of test items in AUTOSAR documents and AUTOSAR meta model.
Rationale:	Consistent specification and representation of test items in AUTOSAR.
Use Case:	[UC_STDT_00019]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04016)

[RS_STDT_00041] Formalized description of BSW Abstract SWS [

Type:	valid
Description:	The standardization template shall be able to publish formalized parts of a general SWS which then are inherited by derived specific SWSs.
Rationale:	Groups of BSW SWSs often share a common set of specification elements. These general specification elements should not be stated multiple times in order to avoid redundancy. Therefore a specific SWS should be able to inherit the general parts from an abstract SWS and focus on those elements that have to be specified in particular.
Use Case:	[UC_STDT_00002],[UC_STDT_00008],[UC_STDT_00015]
Dependencies:	–
Supporting Material:	–

](RS_BRF_04000)

[RS_STDT_00042] Shall provide the ability to define naming conventions for public symbols [

Type:	valid
Description:	The standardization template shall provide the ability to define naming conventions for public symbols. This especially includes requirement ids, module abbreviations, meta data and configuration symbols used in the document of a release
Rationale:	Avoid ambiguities and name clashes inside the specification. Provide a consistent uniform presentation of meta data to the reader of the specification. Allow automatic processing of specification elements.
Use Case:	[UC_STDT_00009]
Dependencies:	–
Supporting Material:	–

](RS_BRF_01024)

3.6 Profiles for Data Exchange Points

[RS_STDT_00101] Description of Data Exchange Point Shall Provide a Human Readable High-Level Overview [

Type:	valid
Description:	The standardization template shall support description of a human readable abstract overview of the data exchange point. This overview is expected to be free text.
Rationale:	Provide brief information to the user about the scope of the profile. Enable the user to decide whether a profile is applicable to an intended data exchange point.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00102] Description of Data Exchange Point Shall Describe Work Product in Methodology [

Type:	valid
Description:	The standardization template shall support description of the work product (i.e. artifact or deliverable) in the methodology which is represented by the description of data exchange point. E.g. via free text in combination with referencing the artifact in the methodology model
Rationale:	It should be clear to which artifacts in the AUTOSAR methodology the profile applies.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00103] Description of Data Exchange Point Shall Describe Intended Use [

Type:	valid
Description:	The standardization template shall support description of the intended use. (e.g. via free text in combination with referencing the work definitions (activities and tasks) in the methodology model, the BSW module or even the list of BSW Requirements)
Rationale:	Define the scope of a profile and the process step in the methodology where it applies.



△

Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	VFB specification - section "VFB Features and Profiles", RS / SRS requirements

](RS_Main_00300)

[RS_STDT_00104] Description of Data Exchange Point Shall Describe Tool and Organization [

Type:	valid
Description:	The standardization template shall support the description of the tools and organizations that are represented by the profile. Examples are: <ul style="list-style-type: none"> • Profile P1 describes the possible output of tool A in version B that was used by organization C in project D • Profile P2 describes a harmonized reference profile of a consumer that is defined by AUTOSAR • Profile P3 describes the harmonized subset of a data exchange point that is supported by a list of tools with specific versions and configurations.
Rationale:	Administrative Data that helps relating profiles with actual tools, organizations or projects
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	ASAM ProjectData

](RS_Main_00300)

[RS_STDT_00105] Description of Data Exchange Point Shall Describe AUTOSAR Revision [

Type:	valid
Description:	The standardization template shall support description of the AUTOSAR revision the rules in the profile relates to.
Rationale:	Specify the AUTOSAR revision that is used as a baseline of the whole profile.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00106] Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of the AUTOSAR Meta-Model [

Type:	valid
Description:	The standardization template shall support description of the subset of the meta-model that is relevant for the data exchange point.
Rationale:	<p>Document which parts of the meta-model are involved at a specific step in the methodology. Helps to avoid interoperability issues:</p> <ul style="list-style-type: none"> • Application of strict XML schema without tailoring • Missing elements/attributes • Structures without content and orphans - undefined behavior • Loss of information • Missing implementation of AUTOSAR feature or modeling pattern • Configuration in ECU Configuration instead of upstream templates, • Multiple possibilities for expressing the same thing
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00107] Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of Model [

Type:	valid
Description:	<p>The standardization template shall support description of the subset of the model that is relevant for the data exchange point.</p> <p>(e.g. by explicitly defining how to navigate through the model)</p>
Rationale:	<p>Distinguish between data that is required for execution of the intended step in the methodology that requires validation and data that is not (yet) used and doesn't require validation.</p> <p>Helps to avoid interoperability issues:</p> <ul style="list-style-type: none"> • Application of strict XML schema without tailoring • Missing elements/attributes • Structures without content and orphans - undefined behavior • Loss of information • Missing implementation of AUTOSAR feature or modeling pattern • Configuration in ECU Configuration instead of upstream templates, • Multiple possibilities for expressing the same thing
Use Case:	[UC_IOAT_00030] in [10]



△

Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00108] Description of Data Exchange Point Shall Describe Relevant Constraints [

Type:	valid
Description:	The standardization template shall support description of constraints that are relevant for the data exchange point.
Rationale:	Reduce risk of interoperability issues that are cause by <ul style="list-style-type: none"> • different interpretation or selection of constraints • not having implemented required AUTOSAR functionality
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00109] Description of Data Exchange Point Shall Describe Relevant Spec Items [

Type:	valid
Description:	The standardization template shall support description of spec items from all template specifications that are relevant for the data exchange point.
Rationale:	Some Spec Items have the character of a checkable constraint. Or the selection of spec items document supported AUTOSAR capabilities. Reduce risk of interoperability issues that are cause by <ul style="list-style-type: none"> • different interpretation or selection of spec items • not having implemented required AUTOSAR functionality
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00110] Description of Data Exchange Point Shall Describe Model Completeness [

Type:	valid
Description:	The standardization template shall support description the completeness of data at a data exchange point. (e.g. by refining the semantics of the lower multiplicity of meta-model references and attributes per exchange point)
Rationale:	If the completeness is not specified for a data exchange point, then consuming tools in a tool chain might require information that is not provided by the producing tool. Avoid interoperability issues that are caused by different interpretation about required model elements and features.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00111] Description of Data Exchange Point Shall Describe Applicability of Default Values [

Type:	valid
Description:	The standardization template shall support describing when a consumer of an AUTOSAR model should apply AUTOSAR defined default values. E.g. "apply on revision update", "never apply", "always apply".
Rationale:	Overcome the weak semantics of the current AUTOSAR defined default values
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00113] Description of Data Exchange Point Shall Describe Limitation of Values of Primitive Attributes [

Type:	valid
Description:	The standardization template shall support the description of restrictions of values of attributes with primitive data types in specific contexts of the specific data exchange point (e.g. limit the values of category or an enumeration)
Rationale:	The consuming tool may only support a subset of the enumeration values or a limited parameter range. E.g. The consuming tool only supports a subset of possible CATEGORY values.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–



△

Supporting Material:	–
-----------------------------	---

](RS_Main_00300)

[RS_STDT_00114] Description of Data Exchange Point Shall Support Severity Levels for Compliance with Individual Rules of the Profile [

Type:	valid
Description:	<p>The standardization template shall support the description of severities of each rule:</p> <p>error model element is considered to be incompatible to the profile</p> <p>warning model element is suspect, but process step can be continued.</p> <p>info when the rule applies, but has no influence on the process step, e.g. "no unit specified for variable x"</p>
Rationale:	Not every inconsistency of a model with the profile shall block the workflow. In an early stage of the development it may be acceptable for the consumer that some data is missing. Nevertheless such issues shall be reported as warnings.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00115] Description of Data Exchange Point Shall Describe Rationales of Decisions [

Type:	valid
Description:	The standardization template shall support description of rationales for any decision that is documented in the specific data exchange point.
Rationale:	Provide additional information on why a specific rule is part of the profile and why the decision was taken. This supports maintainability and helps during discussions about tool incompatibilities.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00116] Description of Data Exchange Point Shall Describe Usage of AUTOSAR Extension Mechanisms [

Type:	valid
Description:	The standardization template shall support description of the usage of AUTOSAR Extension Mechanisms (Category, SDGs) for the specific data exchange point. This includes the textual documentation of the intended use of the extensions including cross references to related spec items and meta classes in newer version of AUTOSAR specifications.
Rationale:	Categories and SDGs are a standard pattern in the AUTOSAR meta model. This provides a solution for project-specific workflows when something is not foreseen in the standard. Nevertheless such extensions are relevant for the interoperability at a data exchange points and it should be possible to validate them with a profile.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00117] AUTOSAR Shall Provide Guidelines for Comparison of Profiles for Data Exchange Points [

Type:	valid
Description:	The standardization template shall define guidelines for comparing / diffing profiles on different abstraction levels. These guidelines provide hints on how to efficiently interpret the diff of two profiles or how to create a normalized version of a composed profile that simplifies comparison.
Rationale:	The language for the description of profiles supports different degrees of formalization. While some parts are intentionally described as free text other parts are optimized for being processed by tools. This allows determining e.g. modifications between different versions of the same tool. Additionally, the rules can limit the effort for analysis of the differences between profiles: e.g. if the comparison on the feature level finds big incompatibilities the comparison on attribute level is no longer necessary.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

](RS_Main_00300)

[RS_STDT_00118] AUTOSAR Shall Provide Guidelines for Compatibility of Profiles for Data Exchange Points [

Type:	valid
Description:	The Standardization template shall define guidelines for checking the compatibility of profiles on different abstraction levels.
Rationale:	<p>Similar to the interface compatibility in programming languages it is not always required to have the same interface. Often it is sufficient if the interfaces are compatible.</p> <p>E.g.: A producer may provide more data than required by a consumer E.g.: Two tools are incompatible if the consumer requires some data that is not provided by the producer</p> <p>Additionally, the rules can limit the effort for analysis of incompatibilities between profiles: e.g.: if the profiles are incompatible on the feature level, the discussion can start on this level before dealing with individual attributes or constraints.</p>
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00120] AUTOSAR Shall Provide Support for Handling of Incomplete Profiles for Data Exchange Points [

Type:	valid
Description:	The Standardization Template Shall define rules for how to compare, check compatibility or check composability of incomplete profiles.
Rationale:	It might not be possible to agree on all aspects of a profile in the context of AUTOSAR. However, it might be possible to agree in the context of smaller groups that further refine the profile provided by AUTOSAR.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00121] AUTOSAR Shall Provide Guidance for Checking Compliance of AUTOSAR Model Against Profiles for Data Exchange Points [

Type:	valid
Description:	The Standardization Template shall define guidance for checking the compliance of AUTOSAR models against a given profile
Rationale:	Check if the exchanged data complies with the agreed contract. Overcome the limitations of checks against the strict XML schema.



△

Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00119] AUTOSAR Shall provide Rules for Composition of Profiles for Data Exchange Points [

Type:	valid
Description:	The Standardization template shall define rules for composing multiple profiles.
Rationale:	Modularization of profiles, combining profiles of multiple consuming tools. e.g.: it is acceptable if one tool requires some data while another considers this data as don't care. e.g.: a conflict occurs if one tool requires some data while another tool excludes it.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00122] AUTOSAR Shall Provide Guidance for Identification of Not Yet Described Aspects within Profiles for Data Exchange Points [

Type:	valid
Description:	The standardization template shall specify guidelines that help to identify not yet described aspects of a data exchange point.
Rationale:	The profile allows for referencing spec items, constraints, meta classes, etc. that are mentioned in AUTOSAR specifications. The guidelines e.g. provides guidance on how to leverage existing information in AUTOSAR specifications in order to find related information (e.g. via upstream mapping, cross references, technical terms, ...)
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	–
Supporting Material:	–

]([RS_Main_00300](#))

[RS_STDT_00123] AUTOSAR Shall Provide Guidance for Consistency of Profiles for Data Exchange Points [

Type:	valid
Description:	The Standardization Template shall define consistency guidelines for profiles for data exchange points
Rationale:	Parts of a profile depend on each other. E.g. statements on default values are only required if the related attributes are relevant. Statements on completeness of meta-classes are only required if the meta-class is reachable from class AUTOSAR via containment references.
Use Case:	[UC_IOAT_00030] in [10]
Dependencies:	—
Supporting Material:	—

]([RS_Main_00300](#))

A Change History

A.1 Change History R4.0.3

A.1.1 Added Use Cases

Number	Heading
[UC_STDT_00001]	Support Application Interfaces
[UC_STDT_00002]	Express Parts of SWS
[UC_STDT_00003]	Standardize ECUCParamdefs
[UC_STDT_00004]	Express predefined Paths
[UC_STDT_00005]	Express PlatformTypes
[UC_STDT_00006]	Express Examples of applied Standards
[UC_STDT_00007]	Support Verification if an implementation adheres to defined Standard
[UC_STDT_00008]	Support reusable Documentation
[UC_STDT_00009]	Define name conventions
[UC_STDT_00010]	Perform Standardization on Levels beyond the AUTOSAR Scope
[UC_STDT_00011]	Derive Objects from Blueprints by manually changing properties
[UC_STDT_00012]	Derive Objects from Blueprints in a completely standardized Way
[UC_STDT_00013]	Integrate compile test
[UC_STDT_00014]	Generate BSW "Standard AUTOSAR Interface" description from model

Table A.1: Added Use Cases in 4.0.3

A.1.2 Added Requirements

Number	Heading
[RS_STDT_00001]	Shall support and explain Blueprints in general
[RS_STDT_00002]	Formalized description of BSW SWS
[RS_STDT_00003]	Shall allow to represent port blueprints
[RS_STDT_00004]	Shall allow to represent <code>shortName</code> patterns
[RS_STDT_00005]	Shall support keywords and keyword abbreviations
[RS_STDT_00006]	Shall be implemented without compatibility problems to existing template
[RS_STDT_00007]	Shall be based on the AUTOSAR schema
[RS_STDT_00008]	Shall provide means to support analyzing the conformity of implementations with the AUTOSAR standards
[RS_STDT_00009]	Shall be able to represent requirements stated in SWS
[RS_STDT_00010]	Shall refer to ECUC parameter definition
[RS_STDT_00011]	Shall be able to standardize components
[RS_STDT_00012]	Shall be able to standardize architecture
[RS_STDT_00013]	Shall be able to express parts of reference paths resp. package hierarchies
[RS_STDT_00014]	Shall be able to express levels of obligation
[RS_STDT_00015]	Shall support different Approaches to derive from Blueprints
[RS_STDT_00016]	Shall be able to express information about the state of model elements
[RS_STDT_00017]	Shall cover the compatibility of blueprints and derived objects
[RS_STDT_00018]	Shall allow to describe the dependencies of APIs (e.g. invocation and callback/polling interfaces)
[RS_STDT_00019]	Shall define the mandatory semantics for a Blueprint
[RS_STDT_00020]	Shall support variants of a VariableDataprototype
[RS_STDT_00021]	Shall support multiple instantiation for an example SWC with PortBlueprint

[RS_STDT_00022]	Means of exchange format between stakeholders for blueprints
[RS_STDT_00023]	Shall be able to standardize Alias Names
[RS_STDT_00024]	Shall be able to standardize Unique Names and Display Names
[RS_STDT_00025]	Shall be able to standardize life cycle states
[RS_STDT_00026]	Shall allow to represent port interface blueprints
[RS_STDT_00027]	Shall allow to evaluate the integrity of Blueprints
[RS_STDT_00028]	Shall allow to generate BSW "Standard AUTOSAR Interface" description from model
[RS_STDT_00029]	Shall be able to represent further Blueprints
[RS_STDT_00030]	Shall allow to standardize package structures

Table A.2: Added Requirements in 4.0.3

A.2 Change History R4.1.1

A.2.1 Added Use Cases

Number	Heading
[UC_STDT_00016]	Manage requirements in AUTOSAR

Table A.3: Added Use Cases in 4.1.1

A.2.2 Added Requirements

Number	Heading
[RS_STDT_00031]	Shall support general specification items
[RS_STDT_00032]	Shall be able to provide Blueprints for Roles and Rights
[RS_STDT_00033]	Shall be able to provide Blueprints for Build Action Manifest
[RS_STDT_00034]	Blueprinting of Implicit Communication Behavior
[RS_STDT_00035]	Shall support blueprinting of keywords
[RS_STDT_00036]	StandardizationTemplate shall specify the representation of requirements in AUTOSAR documents

Table A.4: Added Requirements in 4.1.1

A.3 Change History R4.1.2

A.3.1 Added Use Cases

Number	Heading
[UC_STDT_00017]	Manage specification items in AUTOSAR
[UC_STDT_00018]	Manage constraint items in AUTOSAR

Table A.5: Added Use Cases in 4.1.2

A.3.2 Added Requirements

Number	Heading
[RS_STDT_00037]	StandardizationTemplate shall specify the representation of specification items in AUTOSAR documents
[RS_STDT_00038]	StandardizationTemplate shall specify the representation of constraint items in AUTOSAR documents

Table A.6: Added Requirements in 4.1.2

A.4 Change History R4.1.3

A.4.1 Added Use Cases

none

A.4.2 Added Requirements

none

A.5 Change History R4.2.1

A.5.1 Added Traceables in 4.2.1

Id	Heading
[RS_STDT_00039]	StandardizationTemplate shall specify the representation of test items in AUTOSAR documents
[RS_STDT_00040]	Multiplicity of elements in derived objects
[RS_STDT_00041]	Formalized description of BSW Abstract SWS
[RS_STDT_00042]	Shall provide the ability to define naming conventions for public symbols
[UC_STDT_00019]	Manage test items in AUTOSAR

Table A.7: Added Traceables in 4.2.1

A.5.2 Changed Traceables in 4.2.1

Id	Heading
[RS_STDT_00001]	Shall support and explain Blueprints in general
[RS_STDT_00002]	Formalized description of BSW SWS
[RS_STDT_00003]	Shall allow to represent port blueprints
[RS_STDT_00004]	Shall allow to represent <code>shortName</code> patterns
[RS_STDT_00005]	Shall support keywords and keyword abbreviations
[RS_STDT_00006]	Shall be implemented without compatibility problems to existing template
[RS_STDT_00007]	Shall be based on the AUTOSAR schema
[RS_STDT_00008]	Shall provide means to support analyzing the conformity of implementations with the AUTOSAR standards
[RS_STDT_00009]	Shall be able to represent requirements stated in SWS
[RS_STDT_00010]	Shall refer to ECUC parameter definition

[RS_STDT_00011]	Shall be able to standardize components
[RS_STDT_00012]	Shall be able to standardize architecture
[RS_STDT_00013]	Shall be able to express parts of reference paths resp. package hierarchies
[RS_STDT_00014]	Shall be able to express levels of obligation
[RS_STDT_00015]	Shall support different Approaches to derive from Blueprints
[RS_STDT_00016]	Shall be able to express information about the state of model elements
[RS_STDT_00017]	Shall cover the compatibility of blueprints and derived objects
[RS_STDT_00018]	Shall allow to describe the dependencies of APIs (e.g. invocation and call-back/polling interfaces)
[RS_STDT_00019]	Shall define the mandatory semantics for a Blueprint
[RS_STDT_00020]	Shall support variants of a VariableDataprototype
[RS_STDT_00021]	Shall support multiple instantiation for an example SWC with PortBlueprint
[RS_STDT_00022]	Means of exchange format between stakeholders for blueprints
[RS_STDT_00023]	Shall be able to standardize Alias Names
[RS_STDT_00024]	Shall be able to standardize Unique Names and Display Names
[RS_STDT_00025]	Shall be able to standardize life cycle states
[RS_STDT_00026]	Shall allow to represent port interface blueprints
[RS_STDT_00027]	Shall allow to evaluate the integrity of Blueprints
[RS_STDT_00028]	Shall allow to generate BSW "Standard AUTOSAR Interface" description from model
[RS_STDT_00029]	Shall be able to represent further Blueprints
[RS_STDT_00030]	Shall allow to standardize package structures
[RS_STDT_00031]	Shall support general specification items
[RS_STDT_00032]	Shall be able to provide Blueprints for Roles and Rights
[RS_STDT_00033]	Shall be able to provide Blueprints for Build Action Manifest
[RS_STDT_00034]	Blueprinting of Implicit Communication Behavior
[RS_STDT_00035]	Shall support blueprinting of keywords
[RS_STDT_00036]	StandardizationTemplate shall specify the representation of requirements in AUTOSAR documents
[RS_STDT_00037]	StandardizationTemplate shall specify the representation of specification items in AUTOSAR documents
[RS_STDT_00038]	StandardizationTemplate shall specify the representation of constraint items in AUTOSAR documents
[UC_STDT_00006]	Express Examples of applied Standards

Table A.8: Changed Traceables in 4.2.1

A.5.3 Deleted Traceables in 4.2.1

none

A.6 Change History R4.2.2

A.6.1 Added Traceables in 4.2.2

none

A.6.2 Changed Traceables in 4.2.2

none

A.6.3 Deleted Traceables in 4.2.2

none

A.7 Change History R4.3.0

A.7.1 Added Traceables in 4.3.0

Id	Heading
[RS_STDT_00101]	Description of Data Exchange Point Shall Provide a Human Readable High-Level Overview
[RS_STDT_00102]	Description of Data Exchange Point Shall Describe Work Product in Methodology
[RS_STDT_00103]	Description of Data Exchange Point Shall Describe Intended Use
[RS_STDT_00104]	Description of Data Exchange Point Shall Describe Tool and Organization
[RS_STDT_00105]	Description of Data Exchange Point Shall Describe AUTOSAR Revision
[RS_STDT_00106]	Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of the AUTOSAR Meta-Model
[RS_STDT_00107]	Description of Data Exchange Point Shall Describe Relevant or Excluded Subset of Model
[RS_STDT_00108]	Description of Data Exchange Point Shall Describe Relevant Constraints
[RS_STDT_00109]	Description of Data Exchange Point Shall Describe Relevant Spec Items
[RS_STDT_00110]	Description of Data Exchange Point Shall Describe Model Completeness
[RS_STDT_00111]	Description of Data Exchange Point Shall Describe Applicability of Default Values
[RS_STDT_00113]	Description of Data Exchange Point Shall Describe Limitation of Values of Primitive Attributes
[RS_STDT_00114]	Description of Data Exchange Point Shall Support Severity Levels for Compliance with Individual Rules of the Profile
[RS_STDT_00115]	Description of Data Exchange Point Shall Describe Rationales of Decisions
[RS_STDT_00116]	Description of Data Exchange Point Shall Describe Usage of AUTOSAR Extension Mechanisms
[RS_STDT_00117]	AUTOSAR Shall Provide Guidelines for Comparison of Profiles for Data Exchange Points
[RS_STDT_00118]	AUTOSAR Shall Provide Guidelines for Compatibility of Profiles for Data Exchange Points
[RS_STDT_00119]	AUTOSAR Shall provide Rules for Composition of Profiles for Data Exchange Points
[RS_STDT_00120]	AUTOSAR Shall Provide Support for Handling of Incomplete Profiles for Data Exchange Points
[RS_STDT_00121]	AUTOSAR Shall Provide Guidance for Checking Compliance of AUTOSAR Model Against Profiles for Data Exchange Points
[RS_STDT_00122]	AUTOSAR Shall Provide Guidance for Identification of Not Yet Described Aspects within Profiles for Data Exchange Points

[RS_STDT_00123]	AUTOSAR Shall Provide Guidance for Consistency of Profiles for Data Exchange Points
[RS_STDT_00125]	Support of AUTOSAR Specific Modeling Patterns

Table A.9: Added Traceables in 4.3.0

A.7.2 Changed Traceables in 4.3.0

Id	Heading
[RS_STDT_00007]	Shall be based on the AUTOSAR XML schema

Table A.10: Changed Traceables in 4.3.0

A.7.3 Deleted Traceables in 4.3.0

none

A.8 Change History R4.3.1

A.8.1 Added Traceables in 4.3.1

none

A.8.2 Changed Traceables in 4.3.1

none

A.8.3 Deleted Traceables in 4.3.1

none

B Mentioned Class Tables

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

Class	DataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of any data type.			
Base	<i>ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable</i>			
Subclasses	<i>ApplicationCompositeElementDataPrototype, AutosarDataPrototype</i>			
Attribute	Type	Mul.	Kind	Note
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

Table B.1: DataPrototype

Class	RunnableEntity			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	A RunnableEntity represents the smallest code-fragment that is provided by an AtomicSwComponent Type and are executed under control of the RTE. RunnableEntities are for instance set up to respond to data reception or operation invocation on a server.			
Base	<i>ARObject, AtpClassifier, AtpFeature, AtpStructureElement, ExecutableEntity, Identifiable, Multilanguage Referrable, Referrable</i>			
Attribute	Type	Mul.	Kind	Note
argument (ordered)	RunnableEntity Argument	*	aggr	This represents the formal definition of an argument to a RunnableEntity.
asynchronous ServerCall ResultPoint	AsynchronousServer CallResultPoint	*	aggr	The server call result point admits a runnable to fetch the result of an asynchronous server call. The aggregation of AsynchronousServerCallResultPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes and the variant existence of server call result points in the implementation. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
canBelnvoked Concurrently	Boolean	1	attr	If the value of this attribute is set to "true" the enclosing RunnableEntity can be invoked concurrently (even for one instance of the corresponding AtomicSwComponent Type). This implies that it is the responsibility of the implementation of the RunnableEntity to take care of this form of concurrency. Note that the default value of this attribute is set to "false".
dataRead Access	VariableAccess	*	aggr	RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataReadAccess is subject to variability with the purpose to support the conditional





Class	RunnableEntity			
				<p style="text-align: center;">△</p> <p>existence of sender receiver ports or the variant existence of dataReadAccess in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataReceivePointByArgument	VariableAccess	*	aggr	<p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The result is passed back to the application by means of an argument in the function signature.</p> <p>The aggregation of dataReceivePointByArgument is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data receive points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataReceivePointByValue	VariableAccess	*	aggr	<p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The result is passed back to the application by means of the return value.</p> <p>The aggregation of dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataSendPoint	VariableAccess	*	aggr	<p>RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataSendPoint is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data send points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
dataWriteAccess	VariableAccess	*	aggr	<p>RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>





Class	RunnableEntity			
external TriggeringPoint	ExternalTriggeringPoint	*	aggr	<p>The aggregation of ExternalTriggeringPoint is subject to variability with the purpose to support the conditional existence of trigger ports or the variant existence of external triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=externalTriggeringPoint, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
internal TriggeringPoint	InternalTriggeringPoint	*	aggr	<p>The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
modeAccess Point	ModeAccessPoint	*	aggr	<p>The runnable has a mode access point. The aggregation of ModeAccessPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode access points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=modeAccessPoint, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
modeSwitch Point	ModeSwitchPoint	*	aggr	<p>The runnable has a mode switch point. The aggregation of ModeSwitchPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode switch points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
parameter Access	ParameterAccess	*	aggr	<p>The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a ParameterDataPrototype which may either be local or within a Port Prototype.</p> <p>The aggregation of ParameterAccess is subject to variability with the purpose to support the conditional existence of parameter ports and component local parameters as well as the variant existence of ParameterAccess (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
readLocal Variable	VariableAccess	*	aggr	<p>The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableDataPrototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of readLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit</p>



△

Class	RunnableEntity			
				<p>△</p> <p>InterRunnableVariable or the variant existence of read LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
serverCallPoint	ServerCallPoint	*	aggr	<p>The RunnableEntity has a ServerCallPoint. The aggregation of ServerCallPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes or the variant existence of server call points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
symbol	CIdentifier	1	attr	<p>The symbol describing this RunnableEntity's entry point. This is considered the API of the RunnableEntity and is required during the RTE contract phase.</p>
waitPoint	WaitPoint	*	aggr	<p>The WaitPoint associated with the RunnableEntity.</p>
writtenLocal Variable	VariableAccess	*	aggr	<p>The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>

Table B.2: RunnableEntity