Document Title	Requirements on Basic Software Module Description Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	086
Document Classification	Auxiliary

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.0

Document Change History

Date	Release	Changed by	Description
2016-11-30	4.3.0	AUTOSAR Release Management	• Removed Debugging support requirement [RS_BSWMD_00061]. • Added further rapid prototyping support with [RS_BSWMD_00070] and [RS_BSWMD_00071].
2015-07-31	4.2.2	AUTOSAR Release Management	• Set Debugging support to obsolete [RS_BSWMD_00061].
2014-10-31	4.2.1	AUTOSAR Release Management	• Layout update. • Tracing update.
2013-10-31	4.1.2	AUTOSAR Release Management	• Layout update.

2013-03-15	4.1.1	AUTOSAR Administration	• Provide activating Event API • Specification of resource locking behavior • Make requirements applicable for Application Software Components as well • Provide Rapid Prototyping Support • BSW Service Availability on Partitions • Support configuration of production errors and extended production errors
2011-12-22	4.0.3	AUTOSAR Administration	• Added detailed change history (chapter 5)
2010-02-02	3.1.4	AUTOSAR Administration	Added support for concepts: • AUTOSAR Scheduler harmonization • Triggered Event • Debugging • A2L Generation Support • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Initial Release

Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

Bibliography

[1] Methodology
AUTOSAR_TR_Methodology

[2] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate

[3] Meta Model-generated XML Schema
AUTOSAR_MMOD_XMLSchema

[4] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate

[5] XML Schema Production Rules
AUTOSAR_TPS_XMLSchemaProductionRules

[6] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[7] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral

[8] Requirements on ECU Configuration
AUTOSAR_RS_ECUConfiguration

[9] Requirements on Runtime Environment
AUTOSAR_SRS_RTE

[10] Glossary
AUTOSAR_TR_Glossary

[11] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration

[12] Requirements on Standardization Template
AUTOSAR_RS_StandardizationTemplate

[13] Specification of Memory Mapping
AUTOSAR_SWS_MemoryMapping

[14] Requirements on Software Component Template
AUTOSAR_RS_SoftwareComponentTemplate

[15] Specification of RTE Software
AUTOSAR_SWS_RTE

1 Scope of this document

This document collects the requirements on the Basic Software Module Description template (BSWMD-T).

The main goal of the BSWMD-T is to provide the scheme for the BSWMD. The BSWMD holds information about an implementation of a BSW module or cluster to support the integration on an ECU. Another use-case is to support of the conformance tests of BSW modules.

Three stages of a BSW module can be distinguished in the methodology:

- "BSW module specification" is provided as the standard by AUTOSAR.
  The API may be a specified for all use-cases. The configuration parameter may have a wide range of configuration possibilities. Certain crucial configuration parameters may be missing because of hardware dependencies which can not be described in the specification.
- "BSW module implementation" is one possible implementation of the BSW module specification.
  Only a subset of the specified API may be implemented. Several configuration decisions may have been taken however other configuration parameters are still open for the integrator to choose.
  Vendor-specific configuration parameters may be added in order to allow the configuration of the module's behavior (applicable for all BSW modules), and/or to support the configuration of specific hardware elements, like special register settings (applicable to hardware dependent modules only).
- "Configured BSW module" takes the still open configuration parameters from a concrete BSW module implementation and assigns values to them. The fully configured BSW module can be actually integrated on an ECU.

Each BSW module implementation does come with an own BSW Module Description. It is important to always use the right pair of BSW module implementation and the according BSWMD.

In figure 1.1 the inputs to the activity "Configure ECU" are shown:

- "Collection of Available Software Components" contains references to all the descriptions of Software Components mapped to this specific ECU
- "ECU Extract of System Description" contains the subset of the System Configuration which is relevant for this specific ECU. This includes the communication matrix and the data to signal mapping.
- "BSW Module Description" (requirements are collected in this document).

The output is the "ECU Configuration Description".

Due to the high configurability of some BSW Modules the BSWMD is not capable of capturing all dependencies of the BSW Module's configuration. Therefore it is also possible to update the BSWMD after the BSW Module has been configured and generated to provide more specific information about the configured BSW Module.

One example of a highly configurable BSW Module is the RTE which is almost completely generated and the initial BSWMD - delivered with the unconfigured RTE - can not describe all possible configurations of the RTE in a formal way.

But after the RTE is configured its BSWMD can be updated to contain descriptions of the actual to be generated RTE. This updated BSWMD can then be used to aid in the configuration of other BSW Modules like Os, Debugger, Dlt.

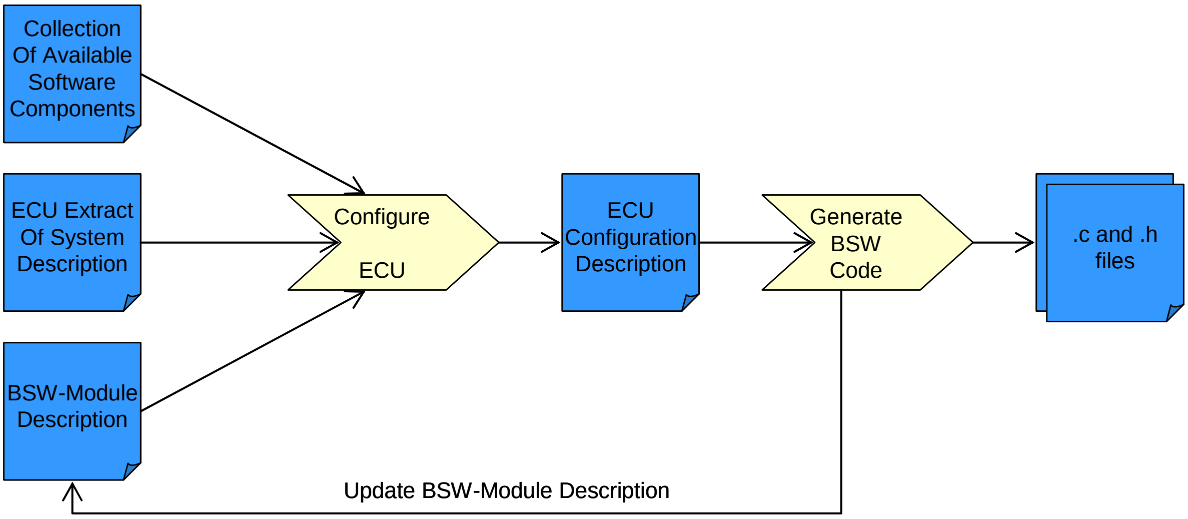
Collection Of Available Software Components
ECU Extract Of System Description
BSW-Module Description
Configure ECU
ECU Configuration Description
Generate BSW Code
.c and .h files
Update BSW-Module Description

Figure 1.1: Overview AUTOSAR Methodology [1]

The BSWMD-template specifies which content the actual Basic Software Module Description (BSWMD) is able to provide. From a technical point of view the template is provided as a document [2] and a XML schema [3] (see also [4] and [5]).

The actual Basic Software Module Descriptions are XML files which conform to the XML schema.

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 ([6]).

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 ([6]).

2 Related Documentationn

2.1 Input Documents

The following input documents have been used in the development of these requirements:

- General Requirements on Basic Software Modules [7]
- Requirements on ECU Configuration [8]
- AUTOSAR RTE Software Requirement Specification [9]
- AUTOSAR Methodology [1]
- AUTOSAR Glossary [10]
- AUTOSAR Generic Structure Template [4]
- AUTOSAR XML Schema Production Rules [5]

2.2 Specification Documents

The requirements collected in this document will be satisfied by:

- Specification of the BSW Module Description Template [2]
  This document implements most of the requirements stated here.
- ECU Configuration Specification [11]
  This document provides specification and guidelines for the creation of Standardized and Vendor-Specific Module Definitions.

2.3 Abbreviations

Abbreviation	***Meaning***
BSW	Basic Software
BSWMD	Basic Software Module Description
BSWMD-T	Basic Software Module Description Template
ECUC	ECU Configuration Values [11]
ECUC Parameter Definition	ECU Configuration Parameter Definition [11]
ECUC-T	ECU Configuration Template [11]
ICS	Implementation Conformance Statement
StMD	Standardized Module Definition [11]
SWC	Software Component Description
SWC-T	Software Component Template
VSMD	Vendor-Specific Module Definition [11]

Table 2.1: Abbrevisations

3 Requirements Tracing

The following table references the requirements specified in [12] and links to the fulfillments of these.

Requirement	Description	Satisfied by
[RS_BRF_00057]	AUTOSAR shall define a memory mapping mechanism	[RS_BSWMD_00031]
[RS_BRF_00206]	AUTOSAR shall support multi-core MCUs	[RS_BSWMD_00066] [RS_BSWMD_00067]
[RS_BRF_01016]	AUTOSAR shall provide a modular design inside software layers	[RS_BSWMD_00039] [RS_BSWMD_00040]
[RS_BRF_01032]	AUTOSAR modules shall provide meta data information	[RS_BSWMD_00010] [RS_BSWMD_00025] [RS_BSWMD_00043]
[RS_BRF_01048]	AUTOSAR module design shall support modules to cooperate in a multitasking environment	[RS_BSWMD_00005] [RS_BSWMD_00011] [RS_BSWMD_00038] [RS_BSWMD_00053] [RS_BSWMD_00054] [RS_BSWMD_00055] [RS_BSWMD_00056] [RS_BSWMD_00057] [RS_BSWMD_00058] [RS_BSWMD_00059] [RS_BSWMD_00060] [RS_BSWMD_00063] [RS_BSWMD_00064] [RS_BSWMD_00066] [RS_BSWMD_00067] [RS_BSWMD_00068]
[RS_BRF_01120]	AUTOSAR shall support re-flashing of configured BSW data	[RS_BSWMD_00013]
[RS_BRF_01136]	AUTOSAR shall support variants of configured BSW data resolved after system start-up	[RS_BSWMD_00013]
[RS_BRF_01160]	AUTOSAR shall support BSW distribution on multi-core MCUs	[RS_BSWMD_00068]
[RS_BRF_01240]	AUTOSAR OS shall support communication between OSApplications	[RS_BSWMD_00066] [RS_BSWMD_00067] [RS_BSWMD_00068]
[RS_BRF_01312]	AUTOSAR RTE shall support procedure-call communication	[RS_BSWMD_00066]

[RS_BRF_01320]	AUTOSAR RTE shall schedule SWC and BSW modules	[RS_BSWMD_00053] [RS_BSWMD_00054] [RS_BSWMD_00055] [RS_BSWMD_00056] [RS_BSWMD_00057] [RS_BSWMD_00058] [RS_BSWMD_00059] [RS_BSWMD_00060]
[RS_BRF_01328]	AUTOSAR RTE shall support scheduling of executable entities on defined events	[RS_BSWMD_00057] [RS_BSWMD_00058] [RS_BSWMD_00059]
[RS_BRF_01360]	AUTOSAR RTE shall support explicit protection mechanisms against concurrent access	[RS_BSWMD_00060] [RS_BSWMD_00064]
[RS_BRF_01368]	AUTOSAR RTE shall support calibration data	[RS_BSWMD_00062]
[RS_BRF_01392]	AUTOSAR RTE shall support a bypass implementation	[RS_BSWMD_00065]
[RS_BRF_01416]	AUTOSAR services shall support standardized handling of non-volatile memory data	[RS_BSWMD_00045]
[RS_BRF_01448]	AUTOSAR services shall support mode and state management	[RS_BSWMD_00054] [RS_BSWMD_00055] [RS_BSWMD_00056]
[RS_BRF_01472]	AUTOSAR shall support modes	[RS_BSWMD_00054] [RS_BSWMD_00055] [RS_BSWMD_00056]
[RS_BRF_01480]	AUTOSAR shall support software component local modes, ECU global modes, and system wide modes	[RS_BSWMD_00054] [RS_BSWMD_00055] [RS_BSWMD_00056]
[RS_BRF_01520]	AUTOSAR RTE shall automatically adapt the runnable management on a mode switch	[RS_BSWMD_00054]
[RS_BRF_02040]	AUTOSAR BSW and RTE shall ensure data consistency	[RS_BSWMD_00060]
[RS_BRF_02072]	AUTOSAR shall provide generic functionality which is in wide use in the automotive domain as libraries	[RS_BSWMD_00037] [RS_BSWMD_00051]
[RS_BRF_02200]	AUTOSAR diagnostic shall provide external access to internal configuration and calibration data	[RS_BSWMD_00062]

[SRS_BSW_00159]	All modules of the AUTOSAR Basic Software shall support a tool based configuration	[RS_BSWMD_00008]

Table 3.1: RequirementsTracing

4 Requirements on BSW Module Description Template

4.1 Published Information

[RS_BSWMD_00043] Support description of common published information ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe the common published information provided by BSW module implementations according to the respective BSW SWS.
Rationale:	Configuration tools SHALL be able to read the common published information of a BSW implementation, as ECU Configuration Values may depend on common published information.
Dependencies:	[RS_BSWMD_00024]
Use Case:	Providing common published information like: Module VERSION, REVISION number or AUTOSAR specification number.
Supporting Material:	–

⌋*(RS_BRF_01032)*

[RS_BSWMD_00024] Support description of module specific published information ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe the module specific published information provided by BSW module implementations according to the respective BSW SWS.
Rationale:	Configuration tools SHALL be able to read the published information of a BSW implementation as ECU Configuration Values may depend on published information.
Dependencies:	[RS_BSWMD_00007], [RS_BSWMD_00043]
Use Case:	Make the value of MEMIF_BROADCAST_ID available to other modules (e.g. to NvM). Make the values of hardware dependent information like: EEPROM-ERASE-TIME or the width of API parameters like EEP-IF-ADDRESSTYPE (uint8, 16, 32) available to other modules (e.g. to MemIf).
Supporting Material:	–

⌋*()*

[RS_BSWMD_00039] Identification of implemented API and functions ⌈

Type:	valid
Description:	Describes which API and functions are actually implemented by the BSW module/cluster.
Rationale:	The specifications of the BSW modules allow implementing only a subset of the specified APIs and functions. Which subset is actually implemented SHALL be described.
Dependencies:	[RS_BSWMD_00040], [RS_BSWMD_00041]
Use Case:	Conformance of a module (cluster) can only be attested for the functionality that a module/cluster actually provides.
Supporting Material:	–

⌋*(RS_BRF_01016)*

[RS_BSWMD_00040] Identification of required API and functions ⌈

Type:	valid
Description:	Describes which API and functions of other modules this implementation requires.
Rationale:	Support the integration by listing the actually needed APIs which are used by this implementation.
Dependencies:	[RS_BSWMD_00039], [RS_BSWMD_00041], [RS_BSWMD_00047]
Use Case:	Check whether the provided API, functions and operation signatures of other modules match the requirements of the BSW module implementation.
Supporting Material:	–

⌋*(RS_BRF_01016)*

[RS_BSWMD_00041] Declaration of the provided API argument data types ⌈

Type:	valid
Description:	Description of the actual data types used by the implementation for API function arguments and ECU Configuration Parameter Definitions which have been left open in the specification documents.
Rationale:	The specifications of the BSW modules in some cases do not fix the data type to be used for the implementation. To allow the integration the actually implemented data types need to be described.
Dependencies:	–
Use Case:	If the BSW SWS does specify an API argument to be either UInt8 or UInt16 the BSWMD template SHALL provide means to describe which type has been used in the actual implementation.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00042] Description of the required API argument data types ⌈

Type:	valid
Description:	Description of the actual data types required by the implementation for API function arguments which have been left open in the specification documents.
Rationale:	The specifications of the BSW modules in some cases do not fix the data type to be used for the implementation. To allow the integration these actually implemented data types need to be described.
Dependencies:	–
Use Case:	If the BSW SWS does specify an API argument to be either UInt8 or UInt16 the BSWMD template SHALL provide means to describe which type is expected in the actual implementation.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00011] Guaranteed execution context of API calls ⌈

Type:	valid
Description:	For API calls to other modules it SHALL be possible to describe whether the call will be executed in interrupt context by the caller.
Rationale:	If both, the caller and the callee specify the context of the call it is possible to detect invalid call chains during ECU configuration activity. If a call is happening in interrupt context it has some restrictions on execution time and available instructions. The RTE Generator needs to know the context of calls from the BSW services to be able to decouple interrupt context from the application SW-Components.
Dependencies:	[RS_BSWMD_00038], [RS_BSWMD_00040]
Use Case:	The Com module expects the notifications from PduR happening in task context, but the PduR just handles the interrupt context which is coming from the CanIf. This is an invalid configuration and shall be detected.
Supporting Material:	–

⌋*(RS_BRF_01048)*

[RS_BSWMD_00038] Required execution context of API calls ⌈

Type:	valid
Description:	A BSWMD template SHALL provide means to define the context for each provided API function in which it shall be invoked.
Rationale:	If both, the caller and the callee specify the context of the call it is possible to detect invalid call chains during ECU configuration activity.
Dependencies:	[RS_BSWMD_00011], [RS_BSWMD_00039]
Use Case:	The Com module expects the notifications from PduR happening in task context, but the PduR just handles the interrupt context which is coming from the CanIf. This is an invalid configuration and shall be detected.
Supporting Material:	–

⌋(*RS_BRF_01048*)

[RS_BSWMD_00010] Compiler version and settings ⌈

Type:	valid
Description:	It SHALL be possible to describe the actual compiler (vendor, version), and its settings, which has been used for object code delivery or which needs to be used for source code delivery.
Rationale:	When BSW is delivered as object code the integrator needs to know how the object code has been compiled. If it is delivered as source code, the code is often provided for specific compilers and versions.
Dependencies:	–
Use Case:	Object code compiled with different compilers may have issues in the stack structure. Therefore the used compiler and its settings have to be described in order to detect such inconsistencies.
Supporting Material:	–

⌋(*RS_BRF_01032*)

[RS_BSWMD_00037] Needed libraries ⌈

Type:	valid
Description:	It SHALL be possible to describe which libraries (vendor and version) have been used for object code deliveries or which need to be included for source code deliveries.
Rationale:	When a BSW module is delivered as object code the integrator needs to know how the object code has to be integrated. If it is delivered as source code, the code may need a specific version of expected libraries only.
Dependencies:	–
Use Case:	If several BSW modules use the same library it only needs to be present on the ECU once. Describe the used libraries and version in order to be able to detect whether the libraries used by several BSW module implementations are not compatible.
Supporting Material:	–

⌋(*RS_BRF_02072*)

[RS_BSWMD_00025] Support for shipment information ⌈

Type:	valid
Description:	The BSWMD template SHALL support the description which files (source, object, documentation) are included in the delivery of the BSW module.
Rationale:	Describe which artifacts are shipped in the delivery of the BSW module.
Dependencies:	[RS_BSWMD_00044]
Use Case:	Check for completeness of the delivered artifacts before integration.
Supporting Material:	–

⌋*(RS_BRF_01032)*

[RS_BSWMD_00014] Support of BSW Module clusters ⌈

Type:	valid
Description:	Support the description of BSW module clusters which implement several BSW modules.
Rationale:	AUTOSAR allows integrating several BSW modules (or even the whole BSW including the AUTOSAR Services) in a single cluster, treating this BSW cluster as one entity. It must be known how the cluster interacts with other modules / clusters in order to integrate it. Tests for clusters must know what parts (operation signatures and configurable functionality) are actually supported by the object under test.
Dependencies:	–
Use Case:	Delivery of complete COM stack in a single implementation. Delivery of the whole AUTOSAR BSW in a single implementation.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00034] ECU Configuration Editor and Generation supported tool version information ⌈

Type:	valid
Description:	It SHALL be possible to describe the supported ECU Configuration editor and generator tool (vendor, version) and its settings.
Rationale:	When a BSW module is delivered the integrator needs to know which editing and generation tools can be used to configure the BSW.
Dependencies:	–
Use Case:	Since the BSW module implementation may need some vendor-specific ECU Configuration Parameter handling it SHALL be possible to state which generator can cope with these extensions.
Supporting Material:	This requirement does not exclude tools not explicitly listed from working with the specific XML file.

⌋*()*

[RS_BSWMD_00013] Describe configuration class of ECU Configuration Parameters ⌈

Type:	valid
Description:	When the actual implementation of a BSW module has the freedom to choose the configuration class (pre-compile, link-time, post-build) it SHALL be possible to describe which alternative has been chosen.
Rationale:	An ECU Configuration Parameter needs to be handled differently depending on its configuration class.
Dependencies:	–
Use Case:	The ECU Configuration editor SHALL be able to only allow changes on the post-build time ECU Configuration Parameters.

Supporting Material:	–

⌋*(RS_BRF_01120, RS_BRF_01136)*

[RS_BSWMD_00033] Pre-configured ECU Configuration Values ⌈

Type:	valid
Description:	The BSWMD template SHALL allow specifying a set of ECU Configuration Values which have been set to fixed values by the implementation.
Rationale:	Pre-configured ECU Configuration Values contain values which cannot be altered by the BSW module integrator, since they are fixed by the implementation. These pre-configured ECU Configuration Values shall be copied into the ECU Configuration Values of the actual BSW module as part of the base module configuration once the module implementation has been chosen [11].
Dependencies:	–
Use Case:	A value may be fixed for different reasons. E.g. all pre-compile parameters are fixed in object code deliveries.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00032] Recommended ECU Configuration Values ⌈

Type:	valid
Description:	The BSWMD template SHALL allow specifying a set of ECU Configuration Values which are recommended by the implementation.
Rationale:	These Recommended ECU Configuration Values may hold the ECU Configuration Values recommended by the implementer and may be copied into the ECU Configuration Values of the BSW module as base, once the BSW module implementation has been chosen. The Recommended ECU Configuration Values are more flexible than default values, since they allow defining several container instances with different ECU Configuration Parameter values in each container [11].
Dependencies:	–
Use Case:	Allow BSW vendors to deliver a partial or complete ECU Configuration Values file of the BSW module together with the implementation. This eases the work of the integrator who only needs to fill in the missing ECU Configuration Values.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00035] Provide Standardized Module Definition ⌈

Type:	valid
Description:	The BSWMD template SHALL allow the specification of the module's Standardized Module Definition.
Rationale:	The Standardized Module Definition is the base for the configuration of the BSW module. Out of the Standardized Module Definition the Vendor-Specific Module Definition is derived.
Dependencies:	[RS_BSWMD_00048]
Use Case:	Provide information which Standardized Module Definition is used with a certain BSW module implementation.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00050] Allow vendor-specific modification of Standardized Module Definition ⌈

Type:	valid
Description:	The BSWMD template SHALL allow modification to the Standardized Module Definition in order to support implementation specific adaptations.
Rationale:	The Standardized Module Definition does specify the superset of possible values per configuration parameter. A certain implementation may restrict the actual applicable characteristics of individual elements in the Standardized Module Definition.
Dependencies:	[RS_BSWMD_00035]
Use Case:	The BlockId of the NvRam manager may either be 8 or 16 bit. The standardized parameter has a min value of 1 and a max value of 65535. An implementation may choose to support only the 8 bit values, so the max value has to be adapted to be 255.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00027] Provide Vendor-Specific Module Definition ⌈

Type:	valid
Description:	The BSWMD template SHALL allow the definition of the module's Vendor-Specific Module Definition.
Rationale:	Additional configuration parameters are necessary because the Standardized Module Definition does not contain all relevant configuration parameters for certain module implementations. Due to the fact that the configuration parameters are hardware dependent the specific implementation does require additional configuration definition. The Vendor-Specific Module Definition specifies which configuration parameters and ranges are actually supported by a concrete implementation of the BSW module.
Dependencies:	[RS_BSWMD_00048]

Use Case:	Vendor-specific configuration parameters may be added in order to allow the configuration of the module's behavior (applicable for all BSW modules), and/or to support the configuration of specific hardware elements, like special register settings (applicable to hardware dependent modules only).
Supporting Material:	For the definition of Vendor-specific parameters the ECU Configuration Parameter Definition template SHALL be used [11].

⌋*()*

[RS_BSWMD_00007] Provide vendor-specific published information ⌈

Type:	valid
Description:	BSWMD template SHALL support the definition of vendor-specific published information.
Rationale:	Vendors may want to publish proprietary information for usage in their tool chain.
Dependencies:	[RS_BSWMD_00048], [RS_BSWMD_00024]
Use Case:	Describe vendor-specific published information on the actual implementation of the module based on the implemented target. Avoid proprietary means to provide such extensions by specifying a standardized way of description.
Supporting Material:	[RS_ECUC_00002] [8]

⌋*()*

[RS_BSWMD_00048] Tagging of Vendor-Specific Module Definition ⌈

Type:	valid
Description:	It SHALL be possible to distinguish between Standardized and Vendor-Specific Module Definitions.
Rationale:	Since it is possible for vendors to add vendor specific ECU Configuration Parameters to the Standardized Module Definition those additions need to be distinguished from the standardized ECU Configuration Parameters.
Dependencies:	[RS_BSWMD_00035], [RS_BSWMD_00027], [RS_BSWMD_00007]
Use Case:	In order to check the Vendor-Specific Module Definition for conformance it needs to be described which ECU Configuration Parameters are standardized and which are vendor-specific.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00047] Modeling of call-chain dependencies between BSW Modules ⌈

Type:	valid
Description:	It SHALL be possible to describe which other API functions are invoked by a function.
Rationale:	Needed when the OS is configured since OS resources must be mapped to the tasks that utilize them.
Dependencies:	[RS_BSWMD_00046]
Use Case:	Derive which OS resources are used when a main function is invoked and this main function is invoking another API function, and so on.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00049] Describe optional and required elements ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe optional and required elements of a BSW module implementation.
Rationale:	Due to the high configurability of the BSW modules - originating from the AUTOSAR specifications - there SHALL be means to describe the actually supported elements of one BSW module implementation.
Dependencies:	–
Use Case:	Statement which optional elements a BSW module implementation supports. Shipment of a BSW module description for production use containing optional elements, which shall be selected later by the integrator. Using a BSW module description which describes the standard including mandatory and optional elements as a reference for conformance checks of other BSW descriptions.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00044] Description of generated artifacts ⌈

Type:	valid
Description:	Support the description of which artifacts a generation tool will create.
Rationale:	The knowledge on which artifacts (header- and c-files, documentation) are generated by the BSW module's generation tool does support integration and build process.
Dependencies:	[RS_BSWMD_00025]
Use Case:	Generate the make-file based on the information from [RS_BSWMD_00025] and the generated artifacts.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00051] Description of libraries ⌈

Type:	valid
Description:	It SHALL be possible to describe libraries and their implementations.
Rationale:	Libraries are used to share code between several users in the BSW and in Application SW-Components. There SHALL be support for the selection and integration of libraries.
Dependencies:	–
Use Case:	Describe the provided APIs of libraries Describe the required APIs of libraries Describe the resource needs of libraries.
Supporting Material:	–

⌋*(RS_BRF_02072)*

[RS_BSWMD_00052] Description of the generated RTE ⌈

Type:	valid
Description:	It shall be possible to describe the attributes of the generated RTE.
Rationale:	The RTE generator is capable of taking many decisions which are affecting the integration of the RTE on an ECU.
Dependencies:	–
Use Case:	Describe the used memory sections of the generated RTE. Describe the generated VFB tracing functions. Describe the resource needs of the actual RTE.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00062] Provide Measurement and Calibration Support ⌈

Type:	valid
Description:	The BSWMDT shall contain a support format for the description of measurement and calibration data, which can be used (together with linker generated information) by external tools to generate the data descriptions required by calibration and measurement tools.
Rationale:	• For the code generated by the RTE, the contained calibration and measurements data are described in several "upstream" artifacts. An external tool shall be able to do the further processing out of a more simple artifact which contains only the relevant information. • An external tool must be able to determine memory addresses of measurement & calibration data. For this, the actual linker symbols of variables and parameters must be made available in the support format. • Information from the ECU configuration (e.g. calibration method by the RTE) must be made available too.
Dependencies:	–

Use Case:	- The RTE generates support data for the measurement and calibration data as part of its own BSWMD, e.g. for data elements in ports. - The RTE (or another tool) generates support data for measurement and calibration data which are declared locally in a module. The BSWMD is updated with the generated support data.
Supporting Material:	–

⌋*(RS_BRF_01368, RS_BRF_02200)*

[RS_BSWMD_00065] Provide Rapid Prototyping Support ⌈

Type:	valid
Description:	The BSWMDT shall contain a support format for the description of rapid prototyping data, which can be used to generate the data descriptions required by rapid prototyping tools.
Rationale:	For the code generated by the RTE, the contained rapid prototyping mechanism shall be described in order to allow rapid prototyping tooling to interact with the RTE.
Dependencies:	–
Use Case:	The RTE generates support data for rapid prototyping as part of its own BSWMD.
Supporting Material:	–

⌋*(RS_BRF_01392)*

[RS_BSWMD_00070] Support post-build hooking tools for rapid prototyping ⌈

Type:	valid
Description:	The BSW module description template shall the support post-build hooking tools for rapid prototyping through the description of the implemented RP memory interface. The RP memory interface consists of unambiguous memory locations at which RP tools can modify the value subsequently used.
Rationale:	Post-build hooking tools needs to identify the correct memory location to insert the hook.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*()*

[RS_BSWMD_00071] Support service based by-passing for rapid prototyping ⌈

Type:	valid

Description:	The BSW module description template shall support the description of rapid prototyping service points and associated instances of events, executables and their access to data. In addition implemented enabling flags implementing conditional execution of executables prepared for rapid prototyping needs to be known.
Rationale:	To support the use of service-based bypass the service points inserted by RTE and the relationship to enabling flags and accessed data needs to be known by the rapid prototyping tooling.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋*()*

[RS_BSWMD_00069] Configuration for production errors and extended production errors ⌈

Type:	valid
Description:	The BSW module description template shall provide the capability to specify the demands on the configuration of Dem for all implemented production errors and extended production errors.
Rationale:	Automatic configuration of Dem.
Dependencies:	–
Use Case:	• service needs for configuration of debounce counters (counting up/down), configuration of limits of these debounce counters etc., • specify whether the Diagnostic modules may request to delete errors. If so, specify how and when errors may be reset.
Supporting Material:	[SWS_BSW_00001]

⌋*()*

4.2 BSW Scheduling

[RS_BSWMD_00053] Cyclic time based scheduling of BSW Main Functions ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe the cyclic time based scheduling requirements of BSW Main Functions.
Rationale:	The RTE Generator creates the scheduling for the whole ECU. Many BSW Modules rely on the cyclic time based call of their main functions in order to fulfill their functionality. The RTE Generator shall be able to implement the cyclic time based call based on the stated requirements.
Dependencies:	–
Use Case:	Call of the function "Com_MainFunctionTx()" from the generated RTE to achieve periodic sending of IPdus.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01320)*

[RS_BSWMD_00054] Mode Switches for BSW modules shall be supported ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe Mode Switches for BSW modules and the requirements on the scheduling.
Rationale:	Conditional scheduling of BSW Main Functions dependent on different operating modes of the ECU. BSW Main Functions are scheduled dependent on modes activated by entering or exiting a mode activated at particular mode transitions
Dependencies:	–
Use Case:	Initialization and finalization phases (Mode provided by the EcuM) Different communication modes (Mode provided by the ComM)
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01320, RS_BRF_01448, RS_BRF_01472, RS_BRF_01480, RS_BRF_01520)*

[RS_BSWMD_00055] Simultaneous Mode transitions ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to specify the required simultaneous switching of a Mode controlling BSW Modules and Application Software Components.
Rationale:	Synchronized behavior during a mode transition controlling AUTOSAR BSW Modules and Application Software Components
Dependencies:	–
Use Case:	ECU global Initialization and finalization phase
Supporting Material:	–

⌋(*RS_BRF_01048*, *RS_BRF_01320*, *RS_BRF_01448*, *RS_BRF_01472*, *RS_BRF_01480*)

[RS_BSWMD_00056] API for Mode switch notification of BSW modules ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe that a particular BswModuleEntity communicates modes.
Rationale:	The code generator for the BSW Scheduler shall generate the Mode Switch API used by the BSW module service as Mode Manager.
Dependencies:	–
Use Case:	EcuM communicates the ECU's operating state to all BSW modules via BSW Scheduler.
Supporting Material:	–

⌋(*RS_BRF_01048*, *RS_BRF_01320*, *RS_BRF_01448*, *RS_BRF_01472*, *RS_BRF_01480*)

[RS_BSWMD_00057] Triggering of BSW Main Functions by Triggered Events ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe the triggering of BSW Main Functions by the occurrence of Triggered Events. Particular BSW Main Functions in BSW dependent from Triggered Events shall be executed after occurrence of the event. The occurrence of the Triggered Event is either reported via API to the BSW Scheduler or by means of the OS (e.g. expiration of an OS Alarm). Restriction: This is only applicable for intra-ECU usage.
Rationale:	Sporadic and non timing based periodic activation of BSW Main Functions in different BSW Modules.
Dependencies:	–
Use Case:	Angle periodic triggering of the ignition for a combustion engine.
Supporting Material:	–

⌋(*RS_BRF_01048*, *RS_BRF_01328*, *RS_BRF_01320*)

[RS_BSWMD_00058] Simultaneous Triggering by Triggered Events ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to specify the required synchronized triggering of Runnable Entities and BSW Main Functions by shared Triggered Events.
Rationale:	Synchronized activation of routines in AUTOSAR BSW modules and Application Software Components.
Dependencies:	–
Use Case:	Angle periodic triggering of the routines in Application Software Components and Complex Device Drivers for a combustion engine.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01328, RS_BRF_01320)*

[RS_BSWMD_00059] API for Triggering BSW modules by Triggered Events ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe that a particular BswModuleEntity raises Triggered Events.
Rationale:	The code generator for the BSW Scheduler shall generate the Trigger API used by the BSW module capturing the source of the Triggered Events.
Dependencies:	–
Use Case:	Angle periodic triggering of the routines in Application Software Components and Complex Device Drivers for a combustion engine.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01328, RS_BRF_01320)*

[RS_BSWMD_00060] Support exclusive areas in BSW Modules and Application Software Components ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to define exclusive areas used by particular BswModuleEntitys to allow priority determination for preventing simultaneous access to shared resources. The exclusive areas shall be defined with a name and the accessing BswModuleEntitys. Exclusive areas shall only protect module internal data.
Rationale:	Decouple the module implementation from applying data consistency mechanisms. The code generator for BSW Scheduler shall provide APIs to enter or exit exclusive areas for BSW Modules.
Dependencies:	–
Use Case:	Provide data consistency for a data buffer shared between a BswSchedulableEntity and a BswInterruptEntity.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01320, RS_BRF_01360, RS_BRF_02040)*

[RS_BSWMD_00063] Allow enabling of providing Activating Bsw Event API ⌈

Type:	valid
Description:	The BSWMD template shall provide means to request, that the Basic Software Scheduler activates the feature to pass the activating BSW-Event to the called Schedulable Entity. The request shall be available per Schedulable Entity.
Rationale:	In case the activating BSW-Event is not required by the Schedulable Entity code it shall not be available and the generated Basic Software Scheduler shall not keep track of the activating BSW-Events for this Schedulable Entity.
Dependencies:	–

Use Case:	A Schedulable Entity is defined to be activated by a "BswTimingEvent" as well as by a "BswDataReceivedEvent". During the execution of the Schedulable Entity the code needs to distinguish which activation source actually triggered the execution.
Supporting Material:	–

⌋(*RS_BRF_01048*)

[RS_BSWMD_00064] Support optional configuration of ExclusiveArea usage within BSWModuleEntities ⌈

Type:	valid
Description:	The BSW module configuration shall support specifying optional configuration information for each implemented BSWModuleEntitiy to describe: • Which ExclusiveAreas are used in a nested way by the entity. • Which other software entities are invoked from within an ExclusiveArea or nested ExclusiveArea.
Rationale:	The additional configuration information can be checked by a tool at configuration time. The goal is to prevent deadlocks by providing warnings to the implementer in case of possible conflicts when resources are shared between different software entities.
Dependencies:	–
Use Case:	• Release of safety critical systems: Deadlocks and some other problems may only occur in unusual situations and cannot all be found by testing. Safe operation of safety critical ASIL systems may require that freedom of deadlocks is proven by static analysis, in addition to other verification and validation methods. • Optimization of resource consumption for locking shared resources: Depending on the distribution of software to tasks and cores, different implementations of ExclusiveAreas are required. By analyzing the dependencies, the most efficient alternative can be chosen. • Integration of 3rd party software: For 3rd party software the source code is typically not available to analyze the way ExclusiveAreas are used. This has to be specified and provided in detail in the configuration description.
Supporting Material:	–

⌋(*RS_BRF_01048*, *RS_BRF_01360*)

[RS_BSWMD_00066] BSW inter-partition client-server communication ⌈

Type:	valid

Description:	The BSWMDT shall provide means to specify required and provided procedure calls for inter-partition and/or inter-core communication within BSW. These shall support both synchronous and asynchronous client-server pattern and shall allow the corresponding BSW Scheduler API to be generated.
Rationale:	Optimize load balancing between the cores w.r.t. BSW services. Increase efficiency of BSW service execution. Enable optimization of multi core SW distribution.
Dependencies:	–
Use Case:	Parallel execution of applications on a multi-core system.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_00206, RS_BRF_01240, RS_BRF_01312)*

[RS_BSWMD_00067] BSW inter-partition sender-receiver communication ⌈

Type:	valid
Description:	The BSWMDT shall provide means to specify required and provided data for inter-partition and/or inter-core communication within BSW. These shall allow the corresponding BSW Scheduler API to be generated.
Rationale:	Optimize load balancing between the cores w.r.t. BSW services. Increase efficiency of BSW service execution. Enable optimization of multi core SW distribution.
Dependencies:	–
Use Case:	Parallel execution of applications on a multi-core system.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_00206, RS_BRF_01240)*

[RS_BSWMD_00068] BSW Service Execution on Local or Remote Partition ⌈

Type:	valid
Description:	If a BSW service is available on a specific partition, the BSWMDT shall allow to create a BSWMD for a given code that supports configurations where the service function is invoked and executed in the same partition as well as configurations where it is invoked in one partition and routed to a different one by the RTE.
Rationale:	Optimize load balancing between the cores w.r.t. BSW services. Increase efficiency of BSW service execution. Enable optimization of multi core SW distribution.
Dependencies:	–
Use Case:	Parallel execution of applications on a multi-core system.
Supporting Material:	–

⌋*(RS_BRF_01048, RS_BRF_01160, RS_BRF_01240)*

4.3 Resources

[RS_BSWMD_00005] Description of the memory needs of the software implementation ⌈

Type:	valid
Description:	The BSWMD template SHALL support the description of the memory needs of an implementation of a software module (Basic Software and Application Software). Also specification of the quality (e.g. estimate, measurement, analysis) of these values SHALL be supported. The memory requirements of the defined memory sections SHALL be described individually.
Rationale:	Resource estimations/measurements are needed to design & configure the ECU.
Dependencies:	[RS_BSWMD_00031]
Use Case:	The ROM utilization of BSW modules delivered as object code is typically fixed and can be stated in the BSWMD. In most cases the memory needs are dependent on the actual ECU Configuration Parameter Values and can only be estimated.
Supporting Material:	–

⌋*(RS_BRF_01048)*

[RS_BSWMD_00031] Description of used memory section names ⌈

Type:	valid
Description:	Support the description of memory section names which have been used while developing/compiling the software module.
Rationale:	With the usage of memory section names it is possible to partition the software into several sections which will be placed into memory sections on the ECU in the ECU Configuration activity.
Dependencies:	[RS_BSWMD_00005]
Use Case:	The ECU State Manager implementation uses the memory section NOINIT to indicate which declared variables SHALL not be initialized during ECU startup. It is up to the ECU Configuration activity to actually map this section in an appropriate memory section on the ECU which satisfies this requirement.
Supporting Material:	Specification of Memory Mapping [13], [RS_ECUC_00068] [8]

⌋*(RS_BRF_00057)*

[RS_BSWMD_00009] Description of peripheral register usage ⌈

Type:	valid
Description:	The BSWMD template SHALL support ECU Configuration tools in determining conflicts between different BSW modules accessing the same peripheral register. In some cases these needs are dependent on the actual ECU Configuration Parameter Values (no formula SHOULD be provided in that case!).

Rationale:	BSW module implementations from different vendors may use conflicting configuration of peripheral registers. When these BSW modules are integrated in the same ECU then the ECU Configuration tool SHOULD detect these conflicts and alert the user.
Dependencies:	–
Use Case:	Two BSW module implementations both writing to the same microcontroller register but using a different setting. Conflict must be identified.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00016] Timing guarantees ⌈

Type:	valid
Description:	The BSWMD template SHALL allow specifying the guaranteed or estimated reaction time of the BSW module functions (main functions and API calls incl. callbacks & ISR).
Rationale:	To be able to do timing analysis of Application SW-Components, the BSW needs to define timing guarantees.
Dependencies:	–
Use Case:	With the knowledge of the guaranteed execution time the design of exclusive area access can be optimized, depending on the duration an interrupt block might be possible.
Supporting Material:	[RS_SWCT_02050] Requirements on Software Component Template [14]

⌋*()*

[RS_BSWMD_00015] Timing requirements ⌈

Type:	valid
Description:	The BSWMD template SHALL allow specifying the timing requirements on functions called in other modules such as callback functions.
Rationale:	To be able to do timing analysis of Application SW-Components, the BSW needs to define timing requirements additional to the Timing guarantees [RS_BSWMD_00016].
Dependencies:	–
Use Case:	In order to fulfill a given timing guarantee the calls to other functions need to be restricted in time.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00030] Publish resource needs for the BSW Scheduler ⌈

Type:	valid
Description:	BSWMD template SHALL provide means to describe the resources used by the implementation which need to be provided and integrated by the BSW Scheduler [15].
Rationale:	The BSW Scheduler is used to abstract the usage of concrete OS mechanism from the abstract concepts. The abstract concepts are describe in the BSW Scheduler specification [15]. The BSWMD template SHALL provide means to describe the needs on the BSW Scheduler from the BSW module implementation. But which actual mechanism is used to fulfill these needs is up to the implementation of the BSW Scheduler.
Dependencies:	[RS_BSWMD_00046]
Use Case:	A BSW module is using an exclusive area access in its implementation and need to describe this usage, but it is up to the BSW Scheduler how this exclusive area access is actually implemented (using global interrupt blocking or an OS resource).
Supporting Material:	–

⌋*()*

[RS_BSWMD_00046] Publish OS resource usage ⌈

Type:	valid
Description:	For every function (main, API, ISR) it SHALL be possible to describe the OS resources used within the function.
Rationale:	To configure the OS correctly the access to OS resources has to be specified for every function. The BSW Scheduler must be able to resolve the task context in which any OS resource may be used.
Dependencies:	[RS_BSWMD_00030], [RS_BSWMD_00047]
Use Case:	Configure the OS with the right OS resource accesses.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00045] Publish resources needed from AUTOSAR Services ⌈

Type:	valid
Description:	If a BSW module needs resources from AUTOSAR Services, the needs have to be described.
Rationale:	To allow the ECU Configuration activity of the AUTOSAR Services, the needs from BSW and Application SW Components have to be captured.
Dependencies:	–
Use Case:	When a BSW module requires some NVRAM space it has to provide a description of the attributes this NVRAM has to have.
Supporting Material:	–

⌋*(RS_BRF_01416)*

[RS_BSWMD_00026] Description of supported hardware ⌈

Type:	valid
Description:	For BSW modules that are hardware dependent (like the drivers) the supported hardware SHALL be described.
Rationale:	Certain software modules can only be integrated on specific hardware.
Dependencies:	The characterization SHOULD be done by referencing the ECU Resource Description.
Use Case:	When the supported hardware is specified a selection of drivers for a certain hardware can be provided.
Supporting Material:	–

⌋*()*

4.4 Requirements on the Template

[RS_BSWMD_00001] Main source of information on BSW Module ECU Configuration activity and integration ⌈

Type:	valid
Description:	The BSWMD template SHALL provide means to describe - or reference to - the information needed for ECU Configuration activity and integration of a BSW module or cluster of BSW modules. This description formats SHALL be used for the ECU Configuration activity and integration along with the relevant BSW SWS documents.
Rationale:	By selecting the BSWMD of a BSW module's implementation, the necessary information for the ECU Configuration activity and integration of that module SHALL be available. When delivering several BSW modules in a cluster the BSWMD template SHALL support the integration of this cluster. However this description format might not formalize all aspects needed for taking integration decisions (e.g. scheduling).
Dependencies:	[RS_BSWMD_00014]
Use Case:	To be able to exchange BSW modules from different vendors only the specified information can be used during the integration.
Supporting Material:	–

⌋*()*

[RS_BSWMD_00008] BSW Module Description SHALL be tool processable ⌈

Type:	valid
Description:	Work products based on the BSWMD template SHALL be readable and processable by tools.
Rationale:	The ECU Configuration activity of an ECU SHALL be supported by tools with the BSWMD as one input.

Dependencies:	–
Use Case:	ECU Configuration activity will have to have tool support. The ICS SHALL be extractable from the BSWMD.
Supporting Material:	–

⌋*(SRS_BSW_00159)*

[RS_BSWMD_00028] Development according to the AUTOSAR Generic Structure Template document ⌈

Type:	valid
Description:	The UML representation of the BSWMD template SHALL be developed according to the AUTOSAR Generic Structure Template.
Rationale:	The experience and tools already available for the AUTOSAR Metamodeling SHALL be reused.
Dependencies:	–
Use Case:	The template for the BSWMD is similar to other templates already done with the AUTOSAR Generic Structure Template.
Supporting Material:	AUTOSAR Generic Structure Template [4]

⌋*()*

[RS_BSWMD_00029] Transformation of BSWMD template modeling according to the AUTOSAR XML Schema Production Rules ⌈

Type:	valid
Description:	The XML representation for the BSWMD template SHALL be derived from its UML representation according to the AUTOSAR XML Schema Production Rules.
Rationale:	The experience and tools already available for the AUTOSAR Modeling SHALL be reused.
Dependencies:	–
Use Case:	The template for the BSWMD is similar to other templates already done with the AUTOSAR Metamodeling Guide.
Supporting Material:	AUTOSAR XML Schema Production Rules [5]

⌋*()*

5 Change History

5.1 Change History for AUTOSAR R4.0.1 against R3.1.5

5.1.1 Added Traceables in R4.0.1

Id	Heading
[RS_BSWMD_00015]	Timing requirements
[RS_BSWMD_00044]	Description of generated artifacts
[RS_BSWMD_00049]	Describe optional and required elements
[RS_BSWMD_00051]	Description of libraries
[RS_BSWMD_00052]	Description of the generated RTE
[RS_BSWMD_00053]	Cyclic time based scheduling of BSW Main Functions
[RS_BSWMD_00054]	Mode Switches for BSW modules shall be supported
[RS_BSWMD_00055]	Simultaneous Mode transitions
[RS_BSWMD_00056]	API for Mode switch notification of BSW modules
[RS_BSWMD_00057]	Triggering of BSW Main Functions by Triggered Events
[RS_BSWMD_00058]	Synchronized Triggering by Triggered Events
[RS_BSWMD_00059]	API for Triggering BSW modules by Triggered Events
[RS_BSWMD_00060]	Support exclusive areas in BSW Modules and Application Software Components
[RS_BSWMD_00061]	Support for Debugging of variables
[RS_BSWMD_00062]	Provide Measurement and Calibration Support

Table 5.1: Added SRS Items in R4.0.1

5.1.2 Changed Traceables in R4.0.1

Id	Heading
[RS_BSWMD_00010]	Compiler version and settings
[RS_BSWMD_00014]	Support of BSW Module clusters
[RS_BSWMD_00025]	Support for shipment information
[RS_BSWMD_00027]	Provide Vendor-Specific Module Definition
[RS_BSWMD_00028]	Development according to the AUTOSAR Generic Structure Template document
[RS_BSWMD_00029]	Transformation of BSWMD modeling according to the AUTOSAR Model Persistence Rules for XML
[RS_BSWMD_00032]	Recommended ECU Configuration Values
[RS_BSWMD_00033]	Pre-configured ECU Configuration Values
[RS_BSWMD_00034]	ECU Configuration Editor and Generation supported tool version information
[RS_BSWMD_00035]	Provide Standardized Module Definition
[RS_BSWMD_00040]	Identification of required API and functions
[RS_BSWMD_00041]	Declaration of the provided API argument data types
[RS_BSWMD_00042]	Description of the required API argument data types
[RS_BSWMD_00043]	Support description of common published information
[RS_BSWMD_00047]	Modeling of call-chain dependencies between BSW Modules
[RS_BSWMD_00048]	Tagging of Vendor-Specific Module Definition
[RS_BSWMD_00050]	Allow vendor-specific modification of Standardized Module Definition

Table 5.2: Changed SRS Items in R4.0.1

5.1.3 Deleted Traceables in 4.0.1

none

5.2 Change History for AUTOSAR R4.1.1 against R4.0.3

5.2.1 Changed Traceables in R4.1.1

Id	Heading
[RS_BSWMD_00005]	Make requirements applicable for Application Software Components as well.
[RS_BSWMD_00031]	Make requirements applicable for Application Software Components as well.

Table 5.3: Added SRS Items in R4.1.1

5.2.2 Added Traceables in R4.1.1

Id	Heading
[RS_BSWMD_00063]	Allow enabling of providing Activating Bsw Event API
[RS_BSWMD_00064]	Support optional configuration of ExclusiveArea usage within BSWModuleEntities
[RS_BSWMD_00065]	Provide Rapid Prototyping Support
[RS_BSWMD_00066]	BSW inter-partition client-server communication
[RS_BSWMD_00067]	BSW inter-partition sender-receiver communication
[RS_BSWMD_00068]	BSW Service Execution on Local or Remote Partition
[RS_BSWMD_00069]	Configuration for production errors and extended production errors

Table 5.4: Added SRS Items in R4.0.4

5.2.3 Deleted Traceables in 4.1.1

none

5.3 Change History for AUTOSAR R4.2.1 against R4.1.3

5.3.1 Added Traceables in 4.2.1

none

5.3.2 Changed Traceables in 4.2.1

Id	Heading
[RS_BSWMD_00005]	Description of the memory needs of the software implementation
[RS_BSWMD_00010]	Compiler version and settings
[RS_BSWMD_00011]	Guaranteed execution context of API calls

[RS_BSWMD_00013]	Describe configuration class of ECU Configuration Parameters
[RS_BSWMD_00025]	Support for shipment information
[RS_BSWMD_00031]	Description of used memory section names
[RS_BSWMD_00037]	Needed libraries
[RS_BSWMD_00038]	Required execution context of API calls
[RS_BSWMD_00039]	Identification of implemented API and functions
[RS_BSWMD_00040]	Identification of required API and functions
[RS_BSWMD_00043]	Support description of common published information
[RS_BSWMD_00045]	Publish resources needed from AUTOSAR Services
[RS_BSWMD_00051]	Description of libraries
[RS_BSWMD_00053]	Cyclic time based scheduling of BSW Main Functions
[RS_BSWMD_00054]	Mode Switches for BSW modules shall be supported
[RS_BSWMD_00055]	Simultaneous Mode transitions
[RS_BSWMD_00056]	API for Mode switch notification of BSW modules
[RS_BSWMD_00057]	Triggering of BSW Main Functions by Triggered Events
[RS_BSWMD_00058]	Simultaneous Triggering by Triggered Events
[RS_BSWMD_00059]	API for Triggering BSW modules by Triggered Events
[RS_BSWMD_00060]	Support exclusive areas in BSW Modules and Application Software Components
[RS_BSWMD_00061]	Support for Debugging of variables
[RS_BSWMD_00062]	Provide Measurement and Calibration Support
[RS_BSWMD_00063]	Allow enabling of providing Activating Bsw Event API
[RS_BSWMD_00064]	Support optional configuration of ExclusiveArea usage within BSWModuleEntities
[RS_BSWMD_00065]	Provide Rapid Prototyping Support
[RS_BSWMD_00066]	BSW inter-partition client-server communication
[RS_BSWMD_00067]	BSW inter-partition sender-receiver communication
[RS_BSWMD_00068]	BSW Service Execution on Local or Remote Partition

Table 5.5: Changed Traceables in 4.2.1

5.3.3 Deleted Traceables in 4.2.1

none

5.4 Change History for AUTOSAR R4.2.2 against R4.2.1

5.4.1 Added Traceables in 4.2.2

none

5.4.2 Changed Traceables in 4.2.2

Id	Heading
[RS_BSWMD_00061]	Support for Debugging of variables

Table 5.6: Changed Traceables in 4.2.2

5.4.3 Deleted Traceables in 4.2.2

none

5.5 Change History for AUTOSAR R4.3.0 against R4.2.2

5.5.1 Added Traceables in 4.3.0

Id	Heading
[RS_BSWMD_00070]	Support post-build hooking tools for rapid prototyping
[RS_BSWMD_00071]	Support service based by-passing for rapid prototyping

Table 5.7: Added Traceables in 4.3.0

5.5.2 Changed Traceables in 4.3.0

Id	Heading
[RS_BSWMD_00029]	Transformation of BSWMD template modeling according to the AUTOSAR XML Schema Production Rules
[RS_BSWMD_00065]	Provide Rapid Prototyping Support

Table 5.8: Changed Traceables in 4.3.0

5.5.3 Deleted Traceables in 4.3.0

Id	Heading
[RS_BSWMD_00061]	Support for Debugging of variables

Table 5.9: Deleted Traceables in 4.3.0