| Document Title | Specification of Basic Software Mode Manager |
|---|---|
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 313 |

| | |
|---|---|
| **Document Status** | published |
| **Part of AUTOSAR Standard** | Classic Platform |
| **Part of Standard Release** | R21-11 |

**Document Change History**

| Date | Release | Changed by | Description |
|---|---|---|---|
| 2021-11-25 | R21-11 | AUTOSAR Release Management | • Adding new mode request for SoAd (BswM_SoAd_SoConModeChg)<br>• Setting BswM_WdgM_RequestPartitionReset and its configuration to obsolete<br>• Editorial Changes |
| 2020-11-30 | R20-11 | AUTOSAR Release Management | • Added new notifications BswMNmStateChangeNotification and BswMSwitchAckNotification; Added new action BswMEthIfStartAllPorts<br>• Added support for Ethernet switch port group switching<br>• Removed support for pretended networking<br>• Reinstated the EcuM<->BswM APIs: BswM_EcuM_CurrentState and BswM_EcuM_RequestedState |

| | | | |
|---|---|---|---|
| 2019-11-28 | R19-11 | AUTOSAR Release Management | • Introduced action list priority parameter<br>• Added configuration options for enabling/disabling SdServiceGroups<br>• Removal of obsolete APIs: BswM_EcuM_CurrentState and BswM_EcuM_RequestedState<br>• Editorial Changes<br>• Changed Document Status from Final to published |
| 2018-10-31 | 4.4.0 | AUTOSAR Release Management | • Reworked handling of EcuM API for shutdown (e.g. BswMEcuMGoDownHaltPoll)<br>• Removed dependency to EcuM-fixed<br>• Changes to BswM_NvM_CurrentJobMode, BswM_ModeType, BswM_UserType and Det error codes |
| 2017-12-08 | 4.3.1 | AUTOSAR Release Management | • Rework of BswM-Com interaction for Pdu group switching and deadline monitoring control. BswM-internal group vectors no longer used<br>• BswMNmIfCarWakeUpIndication reclassified as event request port<br>• Added new dedicated actions for Rte Start/Stop<br>• Editorial changes |
| 2016-11-30 | 4.3.0 | AUTOSAR Release Management | • Added some actions/indications to allow for more BswM interaction with following BSW Modules: EthIf, EcuM<br>• Waiting functionality added using the BswMTimer mode request source<br>• Some mode requests are now modeled using BswMEventRequestPort, instead of BswMModeRequestPort<br>• Editorial changes, increased requirement traceability and minor changes to configuration containers/parameters |

| | | | |
|---|---|---|---|
| 2015-07-31 | 4.2.2 | AUTOSAR Release Management | • Improved specification of service interfaces<br>• Additional functional requirements for BswMPduGroupSwitch action<br>• Added BswMNmIfCarWakeUpIndication as a new BswMModeRequestSource<br>• Deprecated some spec. elements (marked with "obsolete"), editorial changes, increased requirement traceability and minor changes to configuration containers/parameters |
| 2014-10-31 | 4.2.1 | AUTOSAR Release Management | • New API and configuration containers to support EcuM Fixed for Multi Core<br>• Addition of new container for defining mode values: BswMCompuScaleModeValue<br>• New Action BswMFrSMAllSlots for invoking FrSM_AllSlots<br>• New requirements for: Action list execution (SWS_BswM_00223) and Deadline Monitoring (SWS_BswM_00224,00225) |
| 2014-03-31 | 4.1.3 | AUTOSAR Release Management | • Removal of several unnecessary parameter range checks in APIs<br>• J1939 fix: added missing action, missing included header files<br>• Corrections in Figures 1, 2, 3, 5 and 6<br>• Editorial changes |
| 2013-10-31 | 4.1.2 | AUTOSAR Release Management | • Removal of several requirements from Pretended Networking chapter<br>• Addition of new configuration parameters to several Sd related Bswm Actions<br>• Addition of new BswM Mode Request: BswMCanSMIcomIndication<br>• Addition of new BswM Action: BswMRteModeRequest<br>• Editorial changes |

| | | | |
|---|---|---|---|
| 2013-03-15 | 4.1.1 | AUTOSAR Administration | • Extended to support Pretended Networking mode handling<br>• Extended to support Heavy Duty Vehicles and J1939 |
| 2011-12-22 | 4.0.3 | AUTOSAR Administration | • Support of Mode Machine Instances assigned to the SchM<br>• Include of user defined header files<br>• Possibility to provide an initial value for a BswMModeRequestPort |
| 2009-12-18 | 4.0.1 | AUTOSAR Administration | • Include file BswMUserCallout.h added. This user defined header file contains declarations of the call out functions.<br>• Requirement that the BswM module shall perform inter module version checks added<br>• Information added for each configurable action which API to call<br>• Functions BswM_TriggerSlaveRTEStop and BswM_TriggerStartUpPhase2 added to control the start and stop of the RTE on slave cores |
| 2010-02-02 | 3.1.4 | AUTOSAR Administration | • Initial release |

**Disclaimer**

# Table of Contents

# 1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module BSW Mode Manager (BswM).

The BSW Mode Manager is the module that implements the part of the Vehicle Mode Management and Application Mode Management concept that resides in the BSW. Its responsibility is to arbitrate mode requests from application layer SW-Cs or other BSW modules based on simple rules, and perform actions based on the arbitration result.

# 2 Acronyms and abbreviations

| Abbreviation / Acronym | Description |
|---|---|
| BSW | Basic Software |
| BswM | BSW Mode Manager |
| BSWMD | Basic Software Module Description |
| CDD | Complex Driver |
| Dem | Diagnostic Event Manager |
| Det | Default Error Tracer |
| ECU | Electronic Control Unit |
| RTE | Real Time Environment |
| SWC / SW-C | Software Component |
| SWCD | Software Component Description |

**Table 2.1: Table of acronyms and abbreviations**

# 3 Related documentation

## 3.1 Input documents & related standards and norms

[1] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral

[2] Guide to Mode Management
AUTOSAR_EXP_ModeManagementGuide

[3] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral

[4] Requirements on Mode Management
AUTOSAR_SRS_ModeManagement

[5] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager

[6] Specification of RTE Software
AUTOSAR_SWS_RTE

## 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules (see [1]), which is also valid for BSW Mode Manager.

Thus, the specification [1] shall be considered as additional and required specification for BSW Mode Manager.

Information regarding the configuration and usage of the BSW Mode Manager is found in the auxillary document: Guide to Mode Management [2]

# 4 Constraints and assumptions

## 4.1 Limitations

Maximum one instance of the BSW Mode Manager may be used within a partition.

## 4.2 Applicability to car domains

The BSW Mode Manager is applicable to all car domains.

# 5 Dependencies to other modules

The BSW Mode Manager has interfaces to many of the BSW Modules in the AUTOSAR architecture. The majority of these interfaces are however optional and are used based on the needs of each ECU.

The dependencies listed in this chapter are intended to give an overview of some possible interactions between the BswM and other modules. The interactions and modules listed herein should not be considered an exhaustive list of all possibilities.

## 5.1 RTE

The BswM receives mode requests from the SW-Cs via the RTE. Mode Switch Notifications are also propagated to the SW-Cs via the RTE.

## 5.2 EcuM - Flex

EcuM Flex can indicate the state of its wakeup sources to BswM. When ECU Mode Handling is used, BswM can set the state of EcuM - Flex and receives status of certain modes based on the RUN Request Protocol.

## 5.3 WdgM

(This chapter is OBSOLETE)

The WdgM may request partition reset related actions from the BswM via the `BswM_WdgM_RequestPartitionReset` API. The configuration for the WdgM partition reset request is accomplished via the BswMWdgMRequestPartitionReset mode request source.

## 5.4 ComM

Mode Switch Indications originating from the ComM go through the BswM for further propagation to the SW-Cs.

The BswM can request communication modes at the ComM by means of ComMUsers.

## 5.5 COM

The handling of I-PDU Groups in COM is performed by the BswM. As a part of I-PDU group start/stop, it is possible to have the included signal values reset to their corresponding initialization values.

BswM handles the enabling and disabling of deadline monitoring of signals in COM.

BswM can also trigger transmission of an I-PDU.

## 5.6 PduR

The BswM can enable and disable routing groups of I-PDUs in the PDU router.

## 5.7 CanSM

Mode Switch Indications originating from the CanSM go through the BswM for further propagation to the SW-Cs.

## 5.8 LinSM

BswM coordinates switching of LIN Schedule Tables in the LinSM with starting and stopping of the corresponding I-PDU groups in COM.

Mode Switch Indications originating from the LinSM go through the BswM for further propagation to the SW-Cs.

## 5.9 LinTp

The LIN Transport Protocol that is a part of LinIf requests modes from BswM to make sure that the correct LIN Schedule Table is active during LinTp operation.

## 5.10 FrSM

Mode Switch Indications originating from the FrSM go through the BswM for further propagation to the SW-Cs.

The usage of "Single Slot Mode" on FlexRay is controlled by the FrSM by request of BswM. The send capability of the FlexRay stack can be controlled by the BswM via FrSM by calling the API `FrSM_SetEcuPassive`.

## 5.11 EthSM

Mode Switch Indications originating from the EthSM go through the BswM for further propagation to the SW-Cs.

## 5.12 DCM

The DCM performs mode requests to the BswM based on the diagnostic requests it receives.

Example: DCM can request "Disable Normal Communication". During this mode BswM will turn off the corresponding I-PDU groups and NM PDUs.

## 5.13 J1939Dcm

The J1939Dcm reports communication state changes to the BswM for further propagation to the SW-Cs. BswM changes states of J1939Dcm via `J1939Dcm_SetState`.

## 5.14 J1939Nm

The J1939Nm provides a state indication via `BswM_J1939Nm_StateChangeNotification`.

## 5.15 J1939Rm

BswM changes states of J1939Rm via `J1939Rm_SetState`.

## 5.16 NM Interface

BswM will use the `Nm_EnableCommunication` and `Nm_DisableCommunication` to control the NM communication based on the current mode.

Example: In "Disable Normal Communication" mode, the BswM needs to disable NM communication on the corresponding NM channel.

The Nm uses `BswM_Nm_CarWakeUpIndication` to indicate a CarWakeup.

## 5.17 NvM

The NvM module reports the state of its blocks to the BswM via "integration code" registered as NvM callbacks. BswM has actions causing the NvM to read and write all blocks during startup and shutdown.

## 5.18 OS

The features of OS that are required by BswM, are implementation specific.

## 5.19 Sd

The BswM receives status indications from Sd via several exported APIs (see chapter 8.3 for examples). These status indications from Sd can be configured as `BswMModeRequestSource`s.

## 5.20 File structure

The BswM may use interfaces in AUTOSAR BSW modules that are not explicitly defined within this specification.

# 6 Requirements Tracing

The following tables reference the requirements specified in [3] and [4] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document.

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_BSW_00003]** | All software modules shall provide version and identification information | [SWS_BswM_00003] |
| **[SRS_BSW_00101]** | The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function | [SWS_BswM_00002] [SWS_BswM_00043] [SWS_BswM_00044] [SWS_BswM_00261] |
| **[SRS_BSW_00167]** | All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks | [SWS_BswM_00240] [SWS_BswM_00242] [SWS_BswM_00243] [SWS_BswM_00256] [SWS_BswM_CONSTR_00001] [SWS_BswM_CONSTR_00002] |
| **[SRS_BSW_00170]** | The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands | [SWS_BswM_09999] |
| **[SRS_BSW_00301]** | All AUTOSAR Basic Software Modules shall only import the necessary information | [SWS_BswM_00001] [SWS_BswM_00237] [SWS_BswM_00279] |
| **[SRS_BSW_00305]** | Data types naming convention | [SWS_BswM_00041] |
| **[SRS_BSW_00323]** | All AUTOSAR Basic Software Modules shall check passed API parameters for validity | [SWS_BswM_00045] [SWS_BswM_00089] [SWS_BswM_00090] [SWS_BswM_00091] [SWS_BswM_00093] [SWS_BswM_00095] [SWS_BswM_00097] [SWS_BswM_00099] [SWS_BswM_00101] [SWS_BswM_00103] [SWS_BswM_00110] [SWS_BswM_00113] [SWS_BswM_00133] [SWS_BswM_00150] [SWS_BswM_00154] [SWS_BswM_00206] [SWS_BswM_00209] [SWS_BswM_00212] [SWS_BswM_00228] [SWS_BswM_00229] [SWS_BswM_00268] |
| **[SRS_BSW_00336]** | Basic SW module shall be able to shutdown | [SWS_BswM_00119] [SWS_BswM_00120] [SWS_BswM_09999] |
| **[SRS_BSW_00339]** | Reporting of production relevant error status | [SWS_BswM_09999] |
| **[SRS_BSW_00344]** | BSW Modules shall support link-time configuration | [SWS_BswM_00002] |
| **[SRS_BSW_00358]** | The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void | [SWS_BswM_00002] |
| **[SRS_BSW_00384]** | The Basic Software Module specifications shall specify at least in the description which other modules they require | [SWS_BswM_00007] [SWS_BswM_00008] |
| **[SRS_BSW_00385]** | List possible error notifications | [SWS_BswM_00230] |
| **[SRS_BSW_00399]** | Parameter-sets shall be located in a separate segment and shall be loaded after the code | [SWS_BswM_09999] |
| **[SRS_BSW_00400]** | Parameter shall be selected from multiple sets of parameters after code has been loaded and started | [SWS_BswM_09999] |

△

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_BSW_00404]** | BSW Modules shall support post-build configuration | [SWS_BswM_00002] [SWS_BswM_00042] [SWS_BswM_00213] |
| **[SRS_BSW_00405]** | BSW Modules shall support multiple configuration sets | [SWS_BswM_00002] [SWS_BswM_09999] |
| **[SRS_BSW_00406]** | A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called | [SWS_BswM_00077] [SWS_BswM_00078] [SWS_BswM_00079] [SWS_BswM_00080] [SWS_BswM_00081] [SWS_BswM_00082] [SWS_BswM_00083] [SWS_BswM_00084] [SWS_BswM_00086] [SWS_BswM_00109] [SWS_BswM_00112] [SWS_BswM_00132] [SWS_BswM_00134] [SWS_BswM_00149] [SWS_BswM_00153] [SWS_BswM_00159] [SWS_BswM_00205] [SWS_BswM_00208] [SWS_BswM_00211] [SWS_BswM_00227] [SWS_BswM_00266] [SWS_BswM_00267] |
| **[SRS_BSW_00407]** | Each BSW module shall provide a function to read out the version information of a dedicated module implementation | [SWS_BswM_00003] |
| **[SRS_BSW_00409]** | All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration | [SWS_BswM_09999] |
| **[SRS_BSW_00414]** | Init functions shall have a pointer to a configuration structure as single parameter | [SWS_BswM_00002] |
| **[SRS_BSW_00425]** | The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects | [SWS_BswM_00053] |
| **[SRS_BSW_00441]** | Naming convention for type, macro and function | [SWS_BswM_00213] [SWS_BswM_00214] [SWS_BswM_00216] |
| **[SRS_BSW_00452]** | Classification of runtime errors | [SWS_BswM_00238] [SWS_BswM_00239] |
| **[SRS_BSW_00467]** | The init / deinit services shall only be called by BswM or EcuM | [SWS_BswM_00118] |
| **[SRS_ModeMgm_-09116]** | Requesting and releasing the RUN state shall be provided | [SWS_BswM_91004] |
| **[SRS_ModeMgm_-09174]** | The BSW Mode Manager [5] shall support the 'disable normal Communication' | [SWS_BswM_00038] |
| **[SRS_ModeMgm_-09175]** | A configurable Set of Mode dependent enabled and concomitant disabled IPDU groups shall be supported | [SWS_BswM_00038] |
| **[SRS_ModeMgm_-09177]** | The rules of the mode arbitration shall be pre-compile and post-build configurable | [SWS_BswM_00010] [SWS_BswM_00012] [SWS_BswM_00015] [SWS_BswM_00016] [SWS_BswM_00062] [SWS_BswM_00067] [SWS_BswM_00223] [SWS_BswM_00252] [SWS_BswM_00253] [SWS_BswM_00256] |
| **[SRS_ModeMgm_-09178]** | The lists of mode transition specific actions shall be pre-compile and post-build configurable | [SWS_BswM_00017] [SWS_BswM_00018] [SWS_BswM_00019] [SWS_BswM_00037] [SWS_BswM_00054] [SWS_BswM_00055] [SWS_BswM_00062] [SWS_BswM_00067] [SWS_BswM_00147] [SWS_BswM_00223] [SWS_BswM_00271] [SWS_BswM_00272] [SWS_BswM_CONSTR_00001] |

▽

△

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_ModeMgm_-09179]** | The BSW Mode Manager [5] shall provide an Interface to allow Mode Requests of SW-C's | [SWS_BswM_00046] [SWS_BswM_00064] [SWS_BswM_00201] [SWS_BswM_00203] [SWS_BswM_00236] |
| **[SRS_ModeMgm_-09180]** | The BSW Mode Manager [5] shall evaluate the current mode requests | [SWS_BswM_00009] [SWS_BswM_00011] [SWS_BswM_00013] [SWS_BswM_00014] [SWS_BswM_00023] [SWS_BswM_00035] [SWS_BswM_00059] [SWS_BswM_00060] [SWS_BswM_00061] [SWS_BswM_00064] [SWS_BswM_00066] [SWS_BswM_00068] [SWS_BswM_00069] [SWS_BswM_00075] [SWS_BswM_00115] [SWS_BswM_00116] [SWS_BswM_00117] [SWS_BswM_00189] [SWS_BswM_00200] [SWS_BswM_00203] [SWS_BswM_00220] [SWS_BswM_00241] [SWS_BswM_00244] [SWS_BswM_00245] [SWS_BswM_00246] [SWS_BswM_00247] [SWS_BswM_00248] [SWS_BswM_00252] [SWS_BswM_00253] [SWS_BswM_00254] [SWS_BswM_00255] [SWS_BswM_00257] [SWS_BswM_00258] [SWS_BswM_00262] [SWS_BswM_00263] [SWS_BswM_00264] [SWS_BswM_00265] [SWS_BswM_00269] |
| **[SRS_ModeMgm_-09182]** | The BSW Mode Manager [5] shall propagate a performed mode change to all local SW-Cs | [SWS_BswM_00038] [SWS_BswM_00202] [SWS_BswM_00259] |
| **[SRS_ModeMgm_-09183]** | Configurable Mode Activation initiated Reset of Signals to Initial Values shall be supported | [SWS_BswM_00251] |
| **[SRS_ModeMgm_-09184]** | The mode manager shall be able to use a COM interface to activate, respectively deactivate, I-PDU groups | [SWS_BswM_00038] |
| **[SRS_ModeMgm_-09228]** | The BSW Mode Manager [5] shall provide an Interface to allow Mode Requests of BSW Modules | [SWS_BswM_00046] [SWS_BswM_00047] [SWS_BswM_00048] [SWS_BswM_00049] [SWS_BswM_00050] [SWS_BswM_00051] [SWS_BswM_00052] [SWS_BswM_00058] [SWS_BswM_00064] [SWS_BswM_00104] [SWS_BswM_00131] [SWS_BswM_00148] [SWS_BswM_00152] [SWS_BswM_00156] [SWS_BswM_00157] [SWS_BswM_00158] [SWS_BswM_00165] [SWS_BswM_00193] [SWS_BswM_00194] [SWS_BswM_00203] [SWS_BswM_00204] [SWS_BswM_00207] [SWS_BswM_00210] [SWS_BswM_00214] [SWS_BswM_00216] [SWS_BswM_00217] [SWS_BswM_00235] [SWS_BswM_00249] [SWS_BswM_00250] [SWS_BswM_00283] [SWS_BswM_91001] [SWS_BswM_91003] [SWS_BswM_91004] |
| **[SRS_ModeMgm_-09229]** | The mode manager shall be able to make generic, configured callouts of void functions to other BSW modules | [SWS_BswM_00039] [SWS_BswM_00040] |
| **[SRS_ModeMgm_-09230]** | All actions shall only be performed on mode change | [SWS_BswM_00011] [SWS_BswM_00023] [SWS_BswM_00066] [SWS_BswM_00260] |
| **[SRS_ModeMgm_-09240]** | ComM shall notify BswM of any PNC communication state change | [SWS_BswM_00148] |

**Table 6.1: RequirementsTracing**

# 7 Functional specification

This chapter specifies the functional behavior of the BSW Mode Manager. The operation of the BSW Mode Manager basic functionality can be described as two different tasks: Mode Arbitration and Mode Control.

The Mode Arbitration part initiates mode switches resulting from rule-based arbitration of mode requests and mode indications received from SW-Cs or other BSW modules.

The Mode Control part performs the mode switches by execution of action lists containing mode switch operations of other BSW modules.

The BswM should be seen as a mode management framework module in which behavior is completely defined by its configuration.

There may be different ways of implementing the BswM, such as generation of the complete BswM based on the configuration, or as a rule interpreter that parses an encoded configuration at run time.

However, this specification does not intend to specify any implementation details of the BSW Mode Manager. Hence, any examples stated in this document describing design details should be treated as explanatory text and not as requirements.

## 7.1 Mode Arbitration

The Mode Arbitration performed by the BswM is simple and rule-based. The rules used for mode arbitration are specified in the configuration of the BSW Mode Manager module.

The rules are composed of trivial Boolean expressions and the mode arbitration is thus expected to have a low runtime impact.

In order to know what action lists to execute, the BswM is required to detect changes in mode arbitration results from previous rule evaluation. How this is done, and the memory needed to store results, is implementation specific and not described in this document.

### 7.1.1 Arbitration Rules

A rule is a logical expression that is composed of a set of mode request conditions. The rules are evaluated when the input mode requests and mode indications are changed, or during the execution of the BswM main function. The result of the evaluation (True or False) is used to decide about execution of the corresponding mode control Action List.

### 7.1.2 Mode Conditions and Logical Expressions.

The logical expression that comprises a mode arbitration rule can use different operators such as AND, OR, XOR, NOT and NAND. Each term in the expression corresponds to a mode request condition. If the mode condition references a `BswMModeRequestPort`, the condition will verify if a requested or indicated mode is EQUAL or NOT_EQUAL to a certain mode. If the condition references a `BswMEventRequestPort`, the condition will verify if the request port is SET or CLEAR. `BswMEventRequestPort` events requests differ from mode requests in that the requester sends no requested mode/value to the BswM and as such, there is no mode condition for the BswM to evaluate. Rather, there is only the reception of the event for the BswM to evaluate. When the requester sends/calls the event, then the `BswMEventRequestPort` will be in a SET state. The BswM can then later place the `BswMEventRequestPort` into a CLEAR state by executing a `BswMClearEventRequest` action. An example rule with two conditions is shown in Figure 7.1. The rules and the set of available logical operations are defined as a part of the ECU configuration described in chapter 10.2.


**Figure 7.1: Pseudocode representation of an example rule with two conditions.**

**[SWS_BswM_00252]** ⌈When a BswMModeCondition has BswMConditionType = BSWM_EVENT_IS_SET and references a BswMEventRequestPort:

- if the BswMEventRequestPort is in a SET state, then the BswMModeCondition shall evaluate to TRUE
- if the BswMEventRequestPort is in a CLEAR state, then the BswMModeCondition shall evaluate to FALSE

⌋*(SRS_ModeMgm_09180, SRS_ModeMgm_09177)*

**[SWS_BswM_00253]** ⌈When a BswMModeCondition has BswMConditionType = BSWM_EVENT_IS_CLEARED and references a BswMEventRequestPort:

- if the BswMEventRequestPort is in a SET state, then the BswMModeCondition shall evaluate to FALSE
- if the BswMEventRequestPort is in a CLEAR state, then the BswMModeCondition shall evaluate to TRUE

⌋*(SRS_ModeMgm_09180, SRS_ModeMgm_09177)*

**[SWS_BswM_00254]** ⌈When the BswM receives an event on a configured BswMEventRequestPort (e.g. BswM_ComM_InitiateReset is called by the ComM), the BswMEventRequestPort shall be placed in a SET state.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00255]** ⌈When a BswMClearEventRequest action is executed on a BswMEventRequestPort, the BswMEventRequestPort shall be placed in a CLEAR state.⌋*(SRS_ModeMgm_09180)*

### 7.1.3 Requirements of Mode Arbitration

As mentioned above, the BswM accepts mode requests and mode indications as input for the mode arbitration. Mode requests normally originate from the application SW-Cs but may also originate from other BSW modules such as the DCM. Mode indications are always issued by other BSW modules, such as the different bus specific State Managers, the EcuM and the WdgM. In this document, the generic term mode arbitration request corresponds either to a mode indication or to a mode request.

**[SWS_BswM_00009]** ⌈The BswM shall perform mode arbitration based on incoming mode requests.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00035]** ⌈The BswM shall perform mode arbitration based on incoming mode indications.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00278]** ⌈The BswM shall perform mode arbitration based on event requests as well as the clearing of event requests.⌋*()*

Note: All mode arbitration requests (requests and indications) are handled in the same way by the BswM. They are configured by selection of the corresponding mode condition type in the BswMModeRequestSource configuration container.

**[SWS_BswM_00010]** ⌈The BswM shall perform mode arbitration using configured rules.⌋*(SRS_ModeMgm_09177)*

**[SWS_BswM_00012]** ⌈The mode arbitration rules shall be configurable using the module configuration parameters.⌋*(SRS_ModeMgm_09177)*

**[SWS_BswM_00117]** ⌈BswM is not allowed to use results of previous arbitration rule evaluations as input for the logical expressions.⌋*(SRS_ModeMgm_09180)*

Note: Requirement [SWS_BswM_00117] exists to prohibit using the results of rule evaluations as the input to other rule evaluations. It is largely satisfied by the existing structure of the BswM Configuration containers, because the configurable inputs for logical expressions excludes the results of previous rule evaluations.

**[SWS_BswM_00147]** ⌈The action(s) invoked as a result of evaluating a BswM arbitration rule may be called only in the context of an action list.⌋*(SRS_ModeMgm_09178)*

**[SWS_BswM_00189]** ⌈The BswM shall perform mode arbitration based on incoming mode switch notifications.⌋*(SRS_ModeMgm_09180)*

#### 7.1.3.1 Immediate and Deferred Operation

There are two different ways to schedule the processing of the mode arbitration. It is either done immediately within the context of a mode request/indication, or deferred (cyclically) to the main function of the BswM.

An 'immediate' request is executed in the context of the caller. It is the responsibility of the system integrator to ensure that execution of the action list does not jeopardize system performance or consistency.

Especially, if the caller runs (or may run) in interrupt context, the restrictions concerning usage of system functions in interrupt context apply.

The difference between immediate and deferred operation is shown in the sequence diagrams in section 9.1 and 9.2.

**[SWS_BswM_00061]** ⌈A mode arbitration rule may contain any combination of immediate and deferred mode arbitration requests.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00013]** ⌈It shall be possible to configure the BswM to execute the mode arbitration immediately upon a mode arbitration request. This is configured by setting the `BswMRequestProcessing` configuration parameter (within the `BswMModeRequestPort` container) to BSWM_IMMEDIATE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00059]** ⌈Only the mode arbitration rules that use a specific immediate mode condition shall be evaluated by the BswM within the context of that specific mode request/indication.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00014]** ⌈It shall (also) be possible to defer the mode arbitration until the execution of the main function of the BswM. This is configured by setting the `BswMRequestProcessing` configuration parameter (within the `BswMModeRequestPort` container) to BSWM_DEFERRED.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00257]** ⌈It shall be possible to configure the BswM to execute the mode arbitration immediately when an event is set. This is configured by setting the `BswMEventRequestProcessing` configuration parameter (within the `BswMEventRequestPort` container) to BSWM_IMMEDIATE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00258]** ⌈It shall (also) be possible to defer the mode arbitration until the execution of the main function of the BswM. This is configured by setting the `BswMEventRequestProcessing` configuration parameter (within the `BswMEventRequestPort` container) to BSWM_DEFERRED.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00060]** ⌈All rules that use at least one deferred mode condition shall be evaluated during every execution of the main function of BswM.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00068]** ⌈BswM shall postpone mode arbitration requests received during the processing of its main function until it is finished. Any such postponed IMMEDIATE requests shall be processed directly before the BswM main function exits. Any such postponed DEFERRED requests shall be processed in the next subsequent BswM main function.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00069]** ⌈BswM shall postpone mode arbitration requests received during the processing of an IMMEDIATE request until it is finished. Any such postponed IMMEDIATE requests shall be processed directly after the processing of the original IMMEDIATE request. Any such postponed DEFERRED requests shall be processed in the next subsequent BswM main function.⌋*(SRS_ModeMgm_09180)*

The BswM implementation can choose to use protection mechanisms (e.g. Exclusive Area) in order to guarantee that the execution of actions or of the BswM main function does not get interrupted by any other task (for example a higher-priority task).

Terminology clarification for port "updating": Any mode request port has an associated value\state. Updating a port would mean changing its value\state.

**[SWS_BswM_00281]** ⌈BswM shall update the value of an IMMEDIATE mode request port directly before the arbitration actually takes place and not when the mode request port is triggered.⌋*()*

**[SWS_BswM_00282]** ⌈BswM shall update the value of a DEFERRED mode request port when the mode request port is triggered.⌋*()*

### 7.1.4 Arbitration Behavior after Initialization

The behavior of the mode arbitration of BswM after initialization is controlled by the configuration container `BswMModeInitValue`. This parameter may be configured once for each `BswMModeRequestPort` in the configuration.

**[SWS_BswM_00064]** ⌈If the container `BswMModeInitValue` does not exist or the ModeRequest does not already have an initial value, the BswM shall treat the corresponding mode condition as undefined and not use it for mode arbitration until the corresponding mode arbitration request has been updated for the first time.⌋ *(SRS_ModeMgm_09179, SRS_ModeMgm_09228, SRS_ModeMgm_09180)*

**[SWS_BswM_00241]** ⌈BswM shall only arbitrate rules that do not contain any undefined mode conditions within its logical expressions.⌋ *(SRS_ModeMgm_09180)*

The initial value of each `BswMModeRequestPort` after initialization may be controlled by the configuration container `BswMModeInitValue`.

**[SWS_BswM_00203]** ⌈In case `BswMModeInitValue` is defined the BswM shall initialize the corresponding `BswMModeRequestSource` with either the `BswMBswModeInitValue` or the `BswMCompuScaleModeValue` while the BswM is initialized. The BswM shall reject configurations which contain both a `BswMBswModeInitValue` and a `BswMCompuScaleModeValue` for a single `BswMModeInitValue`. This initialization value shall be used for the arbitration rule until the corresponding mode arbitration request has been updated e.g. each call of `BswM_RequestMode` shall update the GenericRequest mode.⌋ *(SRS_ModeMgm_09179, SRS_ModeMgm_09228, SRS_ModeMgm_09180)*

Note: the Rte and SchM modes always have an intial value (see [SRS_Rte_00116])

**[SWS_BswM_00251]** ⌈Upon initialization of the BswM, all `BswMEventRequestPort`s shall be initialized to a CLEAR state.⌋ *(SRS_ModeMgm_09183)*

## 7.2 Mode Control

The Mode Control part of BswM performs all required actions based on the results of the mode arbitration. This is done using Action Lists. An Action List is an ordered list of actions that the BswM executes when triggered by the Mode Arbitration.

The actions in an Action List can be of three types:

1. Calls to other BSW modules or the RTE. A set of pre-defined actions are listed in 7.2.4.
2. Links to other action lists to be included in the execution.
3. Mode arbitration rules. These rules will be evaluated when the corresponding action list is executed. In this way, a hierarchy of rules is obtained.

The BswM is not required to store or react on any BSW module specific return values on its performed actions. Due to this, the different state managers in the BSW indicate their current state to the BswM to be used as input for the mode arbitration.

However, if an error (E_NOT_OK) is returned, the BswM can issue a Det Runtime Error and/or cancel the currently executing action list.

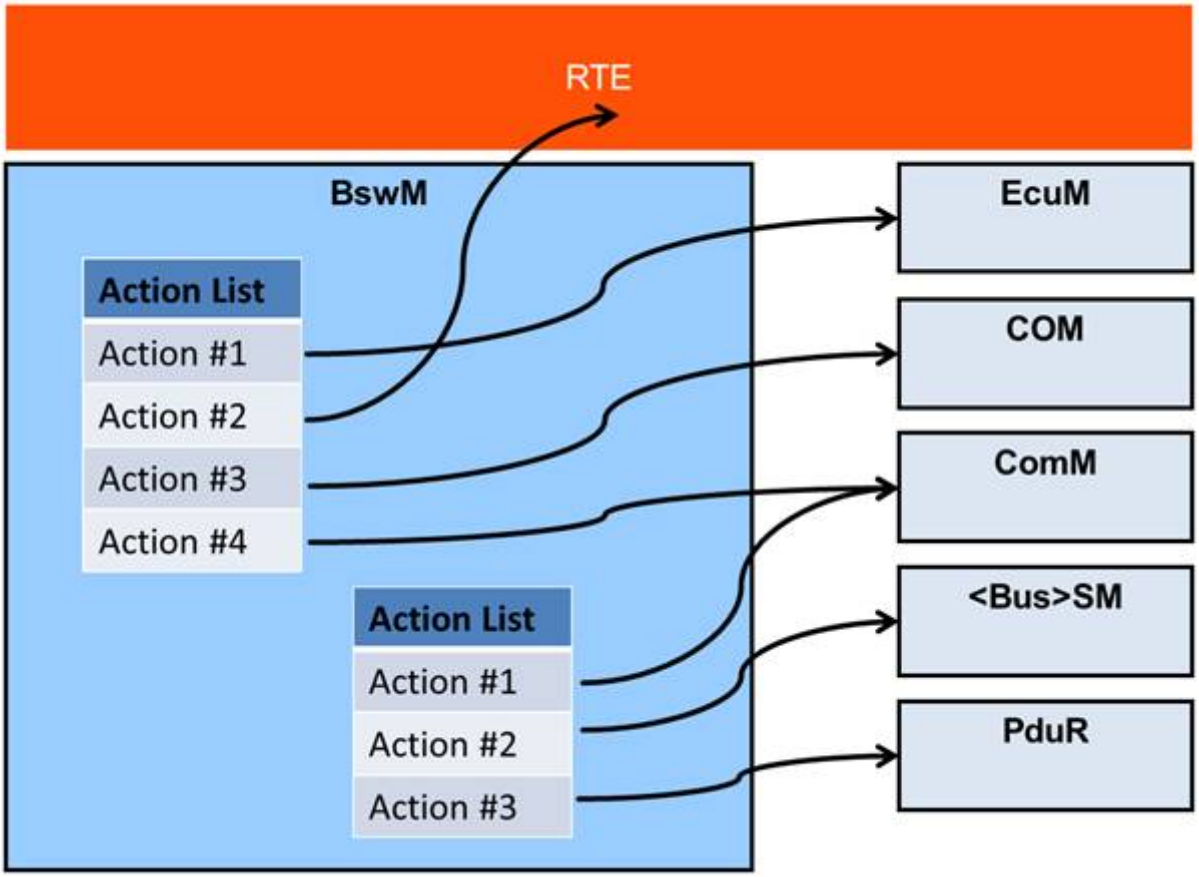


**Figure 7.2: Example showing two action lists**

As shown in Figure 7.2, the BswM may contain multiple Action Lists, and a single Action List can hold multiple actions. To reduce the overall number of action lists, it shall be possible to cascade them. An element of an action list can either be a concrete action or a reference to another action list, or as stated above, a rule to be executed by the mode arbitration. There shall be a flag connected to every action list entry that states its type (action/reference/rule). There shall be no difference between the way a list with concrete actions and the way a list with references or even a mixed list, is activated.

### 7.2.1 Mode Processing Cycle

Figure 7.3 shows the minimal processing cycle for a Mode Request:

1. The Mode Requester SW-C requests mode A through its Sender Port. The RTE distributes the request and the BswM receives it through its Receiver Port.

2. The BswM evaluates its rules either as a result of a received mode arbitration request, or cyclically during the execution of the BswM main function.
3. The corresponding Action List is executed according to the selected execution method (see section "Triggered and Conditional action lists").
4. When executing the Action List, the BswM may issue one or several calls to the RTE Switch API [6] as actions to inform the affected SW-Cs about the arbitration result. Any SW-C, especially the mode requester can register to receive the mode switch indications.

   Note that the mode requester can only receive the mode switch indications from the local BswM; this is true also for requests that originate from a different ECU that is made by a local proxy SW-C.

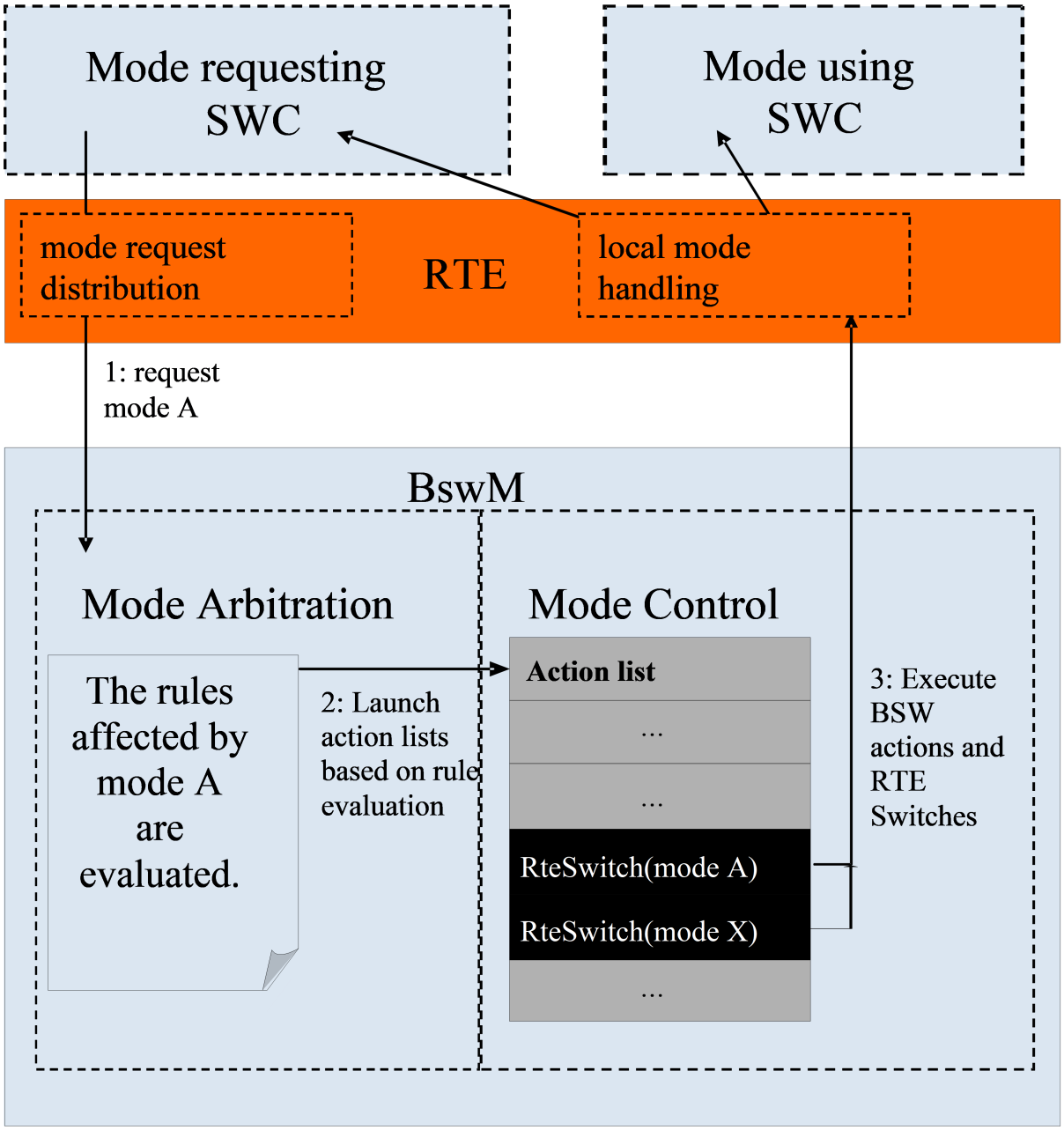


**Figure 7.3: Mode Processing Cycle**

### 7.2.2 Requirements on Mode Control

**[SWS_BswM_00016]** ⌈The BswM shall perform mode control by means of action lists that are executed as a result of rule evaluation in the mode arbitration.⌋*(SRS_ModeMgm_09177)*

**[SWS_BswM_00015]** ⌈For each rule of the mode arbitration, BswM shall be able to execute different action lists based on if the rule evaluates to True or False.⌋*(SRS_ModeMgm_09177)*

**[SWS_BswM_00017]** ⌈An action list comprises a set of actions that BswM shall execute in an ordered manner.⌋*(SRS_ModeMgm_09178)*

**[SWS_BswM_00018]** ⌈An action list may contain links to other action lists that BswM shall include in the execution.⌋*(SRS_ModeMgm_09178)*

**[SWS_BswM_00019]** ⌈An action list may also include links to mode arbitration rules that BswM shall evaluate within the scope of the execution of the current action list.⌋*(SRS_ModeMgm_09178)*

**[SWS_BswM_00067]** ⌈If a rule is included in an action list as specified in [SWS_BswM_00019], any action list execution resulting from that evaluation shall be executed by BswM before it continues to execute the original action list.⌋*(SRS_ModeMgm_09177, SRS_ModeMgm_09178)*

**[SWS_BswM_00037]** ⌈If cascaded action lists are used (i.e. using references to other rules or action lists) the action list structure may contain up to seven (7) hierarchic levels.

Note: The purpose of this limit is to make testing of BswM implementations and generator tools possible. The limit must be checked by the generator tool.⌋*(SRS_ModeMgm_09178)*

**[SWS_BswM_00062]** ⌈Action lists associated with rules evaluated in the context of the mode arbitration request shall be executed by BswM immediately when triggered by the mode arbitration, and not be deferred to the main function execution.

Rationale: This allows very short latencies on mode requests when necessary.⌋*(SRS_ModeMgm_09177, SRS_ModeMgm_09178)*

**[SWS_BswM_00223]** ⌈If a top-level action list is triggered by multiple rules during mode arbitration, this shall result in a single trigger to execute the action list during mode control.⌋*(SRS_ModeMgm_09177, SRS_ModeMgm_09178)*

A top-level action list is an action list which is directly executed by a top-level rule (i.e. a rule which is not nested within an action list), and which is not nested within another action list. [SWS_BswM_00223] only applies to top-level action lists. [SWS_BswM_00223] does not apply to nested rules and nested action lists, since their order within the parent action list is user-defined and should be respected.

**[SWS_BswM_00275]** ⌈If more than one top-level Action List is to be executed during Mode Control, the order of execution shall start with the highest BswMActionListPriority and continue to the lowest. In cases where `BswMActionListPriority` are the same, the order of execution is arbitrary.⌋*()*

**[SWS_BswM_00276]** ⌈The `BswMActionListPriority` shall be ignored for action lists which are not top-level action lists.⌋*()*

**[SWS_BswM_00277]** ⌈An Action List with no `BswMActionListPriority` shall be interpreted as having a `BswMActionListPriority` equal to 0.⌋*()*

**[SWS_BswM_CONSTR_00001]** ⌈The BswM shall reject configurations where a BswMActionList contains BswMActionListItems with same-valued BswMActionListItemIndexes.⌋*(SRS_ModeMgm_09178, SRS_BSW_00167)*

**[SWS_BswM_00260]** ⌈When executing a BswMActionList: the BswM shall start with the BswMActionListItem that has the lowest-valued BswMActionListItemIndex. Subsequent BswMActionListItems shall be executed in increasing order of their BswMActionListItemIndex.⌋*(SRS_ModeMgm_09230)*

Within an action list, the configured BswMActionListItemIndexes do not necessarily need to be contiguous or zero-based. The BswM will start execution of the action list item with the lowest index, and continue to the one with the highest. If the indexes have "gaps" (i.e. are not contiguous), these gaps will simply be ignored.

Because the action list is an ordered list, it is not allowed to configure BswMActionListItemIndexes of the same value within the context of an action list.

### 7.2.3 Triggered and Conditional action lists

There are two ways that an action list may be executed based on evaluation of rules. Either it is executed every time the rule is evaluated with the corresponding result, or only when the evaluation result has changed from the previous evaluation. The execution method for an action list is configured using the BswMActionListExecution parameter (within the BswMActionList container).

However, for a nested action list which is not directly referenced by rules, the BswMActionListExecution parameter (e.g. BSWM_CONDITION or BSWM_TRIGGER) has no meaning, and will have no effect on the way that the nested action list is executed. Such a nested action list (i.e. not directly referenced by rules) is accordingly executed whenever its parent action list is executed.

**[SWS_BswM_00011]** ⌈If a True action list is configured for triggered execution, the BswM shall only execute it when the evaluation of the corresponding rule changes from False to True.⌋*(SRS_ModeMgm_09180, SRS_ModeMgm_09230)*

**[SWS_BswM_00023]** ⌈If a False action list is configured for triggered execution the BswM shall only execute it when the evaluation of the corresponding rule changes from True to False.⌋*(SRS_ModeMgm_09180, SRS_ModeMgm_09230)*

**[SWS_BswM_00115]** ⌈If a True action list is configured for conditional execution, the BswM shall execute it every time the corresponding rule is evaluated to True.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00116]** ⌈If a False action list is configured for conditional execution, the BswM shall execute it every time the corresponding rule is evaluated to False.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00055]** ⌈The BswM shall abort the execution of an action list if an action returns E_NOT_OK and the corresponding `BswMAbortOnFail` configuration parameter is set to "true".⌋*(SRS_ModeMgm_09178)*

### 7.2.4 Available Actions

The set of actions that are available to use in an action list is predefined. The reason for this is to ease ECU configuration and generation of BswM configuration code.

**[SWS_BswM_00038]** ⌈BswM shall be able to execute the predefined actions defined by configuration container `BswMAvailableActions`.⌋*(SRS_ModeMgm_09175, SRS_ModeMgm_09174, SRS_ModeMgm_09182, SRS_ModeMgm_09184)*

**[SWS_BswM_00039]** ⌈The BswM shall be able to call any function in the AUTOSAR BSW even though it is not among the standardized actions defined in `BswMAvailableActions`.⌋*(SRS_ModeMgm_09229)*

**[SWS_BswM_00040]** ⌈The BswM shall be able to call user defined functions.⌋*(SRS_ModeMgm_09229)*

**[SWS_BswM_00054]** ⌈The parameters of the user defined functions, and their values, shall be defined at ECU configuration time using the `BswMUserCallout` configuration container.⌋*(SRS_ModeMgm_09178)*

### 7.2.5 Behavior of Mode Control after Initialization

The behavior of the Mode Control after initialization of the BswM is configured by the `BswMRuleInitState` parameter (within the `BswMRule` container). It defines the "previous evaluation result" to be used when deciding on what action list to execute after the first evaluation of a rule after initialization. The configuration parameter `BswMActionListExecution` (within the `BswMActionList` container) also affects the action list execution after initialization.

**[SWS_BswM_00066]** ⌈The BswM shall act according to what is stated in Table 7.1 when a rule is evaluated for the first time after initialization.

| BswMRuleInitState | BswMActionListExecution | Rule evaluated to true | Rule evaluated to false |
|---|---|---|---|
| BSWM_UNDEFINED | BSWM_TRIGGER | Execute "true" action list | Execute "false" action list |
| BSWM_TRUE | BSWM_TRIGGER | Do nothing | Execute "false" action list |
| BSWM_FALSE | BSWM_TRIGGER | Execute "true" action list | Do nothing |
| BSWM_UNDEFINED | BSWM_CONDITION | Execute "true" action list | Execute "false" action list |
| BSWM_TRUE | BSWM_CONDITION | Execute "true" action list | Execute "false" action list |
| BSWM_FALSE | BSWM_CONDITION | Execute "true" action list | Execute "false" action list |

**Table 7.1: Usage of the BswMRuleInitState configuration parameter**

Note: The "true" and "false" action lists are optional for each rule.⌋*(SRS_ModeMgm_09180, SRS_ModeMgm_09230)*

## 7.3 Waiting Functionality

Sometimes it is necessary to delay specific actions or wait for further mode controls. For this reason a Timer handling was added to the BswM.

A Timer consists always of a BswMTimer as BswMModeRequestSource and corresponding actions (see BswMTimerControl) controlling this BswMTimer i.e. the timer can only be controlled in the context of the action BswMTimerControl -> BswMModeRequestSource/BswMTimer. The value of the BswMTimer (e.g. BSWM_TIMER_STOPPED, BSWM_TIMER_STARTED, BSWM_TIMER_EXPIRED) can be evaluated by other rules configured in the BswM, in order to trigger action lists. There is no external interface to control or manipulate the timer.

**[SWS_BswM_00261]** ⌈Each BswMTimer shall be stopped (BSWM_TIMER_STOPPED) during initialization.⌋*(SRS_BSW_00101)*

**[SWS_BswM_00262]** ⌈The action BswMTimerAction BSWM_TIMER_START shall reload the referenced BswMTimer (via BswMTimerRef) with the corresponding timer value (refer BswMTimerValue) and change the mode of the timer to BSWM_TIMER_STARTED.⌋*(SRS_ModeMgm_09180)*

Note: The timer can only reload by the BswMTimerAction action (no automatic reload possible).

**[SWS_BswM_00263]** ⌈Each BswMTimer in mode BSWM_TIMER_STARTED shall decrement the timer during the BswM_MainFunction (by the cycle time of the BswM_MainFunction).⌋*(SRS_ModeMgm_09180)*

Note: The BswMTimer resolution is a multiple of the BswM_MainFunction cycle. Also, the accuracy of the BswMTimer depends on the accuracy of the BswM_MainFunction.

**[SWS_BswM_00264]** ⌈In case a BswMTimer which is in mode BSWM_TIMER_STARTED expires, its mode shall be changed to BSWM_TIMER_EXPIRED, and then the BswMTimer mode shall be arbitrated in the same BswM_MainFunction cycle.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00265]** ⌈The action BswMTimerAction BSWM_TIMER_STOP shall stop the referenced BswMTimer (via BswMTimerRef) immediately and change its' mode to BSWM_TIMER_STOPPED.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00220]** ⌈The BswMRequestProcessing (e.g. IMMEDIATE, DEFERRED) configuration associated with a BswMTimer shall be ignored by the BswM. The BswM shall treat the processing of a BswMTimer always as DEFERRED; the BswMTimer is arbitrated during the BswM main function.⌋*(SRS_ModeMgm_09180)*

Note: A BswMTimer in mode BSWM_TIMER_EXPIRED will not automatically be set to BSWM_TIMER_STOPPED by the BswM. An action will need to be configured by the user in order to transition the BswMTimer from BSWM_TIMER_EXPIRED into another mode. If there is no action configured to transition the BswMTimer out of

mode BSWM_TIMER_EXPIRED, then the BswMTimer will continue to be arbitrated as BSWM_TIMER_EXPIRED in the following BswM main function cycles.

## 7.4 Multi Partition Support

For multiple BswM instances, each BswM instance will generate its own separate service component description based on its own config set. The integrator will need to allocate these separate service components to the corresponding partition.

BswM is present in each partition with a partition specific configuration (separate instance of BswMConfig per partition). The contained action lists are executed partition locally.

## 7.5 Error classification

Section 7.2 "Error Handling" of the document "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below

### 7.5.1 Development Errors

**[SWS_BswM_00230]** ⌈

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| A service was called prior to initialization | BSWM_E_UNINIT | 0x01 |
| A null pointer was passed as an argument | BSWM_E_NULL_POINTER | 0x02 |
| A parameter was invalid (unspecific) | BSWM_E_PARAM_INVALID | 0x03 |
| A requesting user was out of range | BSWM_E_REQ_USER_OUT_OF_RANGE | 0x04 |
| A requested mode was out of range | BSWM_E_REQ_MODE_OUT_OF_RANGE | 0x05 |
| The provided configuration is inconsistent | BSWM_E_PARAM_CONFIG | 0x06 |
| A parameter pointer was invalid | BSWM_E_PARAM_POINTER | 0x07 |
| Invalid configuration set selection | BSWM_E_INIT_FAILED | 0x08 |

⌋(*SRS_BSW_00385*)

### 7.5.2 Runtime Errors

**[SWS_BswM_00238]** ⌈

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| An action returned E_NOT_OK | BSWM_E_ACTION_FAILED | 0x80..0xFF (as configured in BswMReportFailRuntimeErrorId) |

⌋(*SRS_BSW_00452*)

**[SWS_BswM_00239]** ⌈If `BswMReportFailRuntimeErrorId` is configured for a `BswMActionListItem`, then the BswM shall report a BSWM_E_ACTION_FAILED Runtime Error to Det if the action returns E_NOT_OK. The ErrorId reported in the BSWM_E_ACTION_FAILED Runtime Error is given by the value configured in `BswMReportFailRuntimeErrorId`.⌋(*SRS_BSW_00452*)

Since the calling context of the action depends on the configuration (e.g. DEFERRED or IMMEDIATE), the ApiId reported in the BSWM_E_ACTION_FAILED Runtime Error is not defined in this specification and may be implementation specific.

The BSWM_E_ACTION_FAILED Runtime Error represents a range of ErrorId values. The range of values is restricted to the values given in the table for Runtime Error Types.

**[SWS_BswM_00240]** ⌈The BswM shall reject configurations where a `BswMReportFailRuntimeErrorId` does not lie within the range of values given for BSWM_E_ACTION_FAILED in the Runtime Error Types table.⌋(*SRS_BSW_00167*)

### 7.5.3 Transient Faults

There are no transient faults.

### 7.5.4 Production Errors

There are no production errors.

### 7.5.5 Extended Production Errors

There are no extended production errors.

## 7.6 BswM Interfaces and Ports

This chapter specifies the AUTOSAR Interfaces and Ports that are provided by the Basic Software Mode Manager. Note that ports on both sides of the RTE are required: The SW-C description of the Basic Software Mode Manager services will define the

ports below the RTE. Each AUTOSAR SW-C, which uses the services, must contain service ports in its own SW-C description. These ports are typed with the same interfaces and have to be connected to the ports of the Basic Software Mode Manager, so that the RTE can generate the appropriate IDs and the required symbols.

SW-Cs request modes from the BSW Mode Manager. To that end, they provide a Sender Port that has a special Sender/Receiver Interface (Mode Request Interface) with one data element. The corresponding Receiver Port at the BSW Mode Manager is described in Chapter 7.6.1. The data element's type has the same values as the Mode Declarations in the Mode Declaration Group of the corresponding mode (since the ImplementationDataType of the data element is mapped to the ModeDeclaration Group).

The same SW-C that requests a mode may also be a mode user because it may also need to know the arbitration result of the BSW Mode Manager. The SW-C has a Mode Switch Port, which is a R-Port with a Mode Switch Interface with one data element. This data element's type is then the Mode Declaration Group itself. In addition, other SW-Cs that do not request modes, but depend on them, have such a Mode Switch Port. See Chapter 7.6.3 for a detailed description of the interface to mode users. Note that the BSW Mode Manager also needs a Mode Switch R-Port if it needs to know the current mode in addition to the requested one in its decisions.

Mode Notifications are dispatched by the RTE when a Mode Manager switches the corresponding mode. To do that, the BSW Mode Manager has a Provided type Mode Switch Port that the SW-Cs can connect to. See Chapter 7.6.2 for a detailed description of Mode Switch Ports.

In the context of the requesting SW-C, a Mode Request Port (Sender/Receiver) is defined. The configuration of BSW Mode Manager references this port definition. Let us assume that the SW-C defines an Application Mode `AppModeType`, a corresponding `AppModeRequestType` and an `AppModeTypeMap` that maps the two types to each other:

```
ModeDeclarationGroup AppModeType {
    { APP_MODE_A, APP_MODE_B, APP_MODE_C }
    initialMode = APP_MODE_A;
};


ImplementationDataType AppModeRequestType {
    lowerLimit = 0;
    upperLimit = 2;
};


ModeRequestTypeMap AppModeTypeMap {
    modeGroup = AppModeType;
    implementationDataType = AppModeRequestType;
```

```
};
```

In the context of the SW-C, two Interfaces are defined: the `AppModeRequestInterface` of Sender/Receiver type where the SW-C is sender, and the `AppModeInterface` of Mode Switch type where the SW-C can have P-Ports and R-Ports depending upon the usage:

Figure 7.4 shows how the ports of the application SW-Cs connect to the service ports of the BSW Mode Manager. The Application Mode Manager SW-C has a Mode Request Port and a Mode Switch R-port (named `modeNotificationPort` to distinguish it from the Mode Switch P-ports). The first port is to request changes in its application mode, the latter to receive notifications when the BswM has performed the mode change. The Mode Request Port of the Application Mode Manager (`modeRequestPort0`) connects to the corresponding Mode Request Port of the BSW Mode Manager. Since this is normal Sender/Receiver communication, the Application Mode Manager may connect to multiple BSW Mode Managers even on remote ECUs.

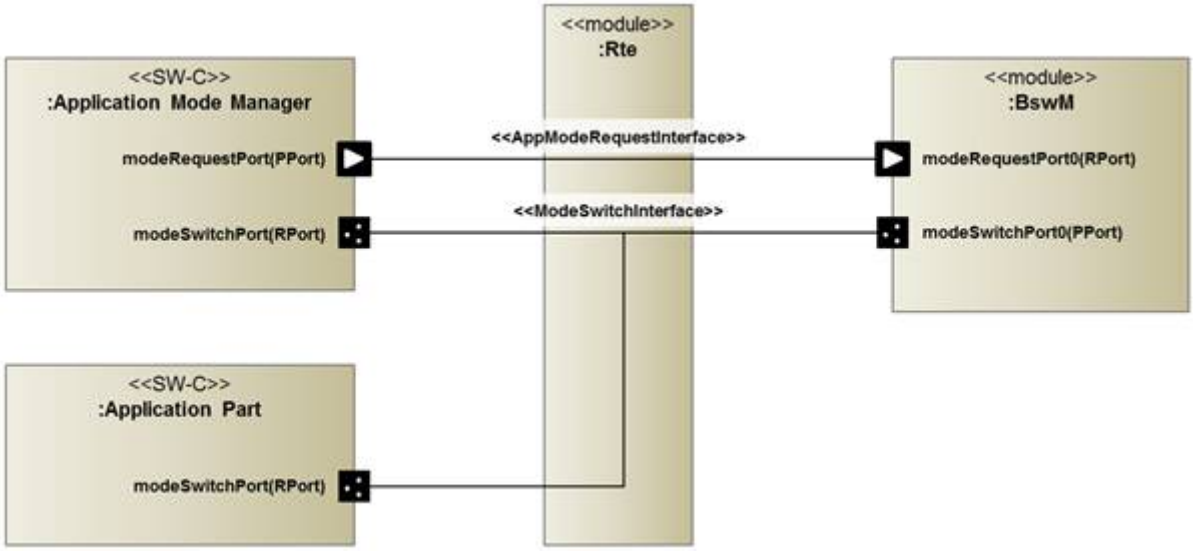


**Figure 7.4: Connections between Application Mode Manager, Application Parts and the BSW Mode Manager**

In order to switch the application mode, the BSW Mode Manager has a Mode Switch Port (`modeSwitchPort_{Name}`) that is implemented by the local RTE.

When the RTE performs the mode switch, it informs all connected entities (BSW Modules or SW-Cs) that are connected via Mode Switch R-Ports to the providing port. The following example presents the Application Mode Manager, the other mode-dependent Application Part and the BSW Mode Manager itself (Note that it's named `modeNotificationPort_{Name}` but the port type is Mode Switch Port). All of these connections are also local.

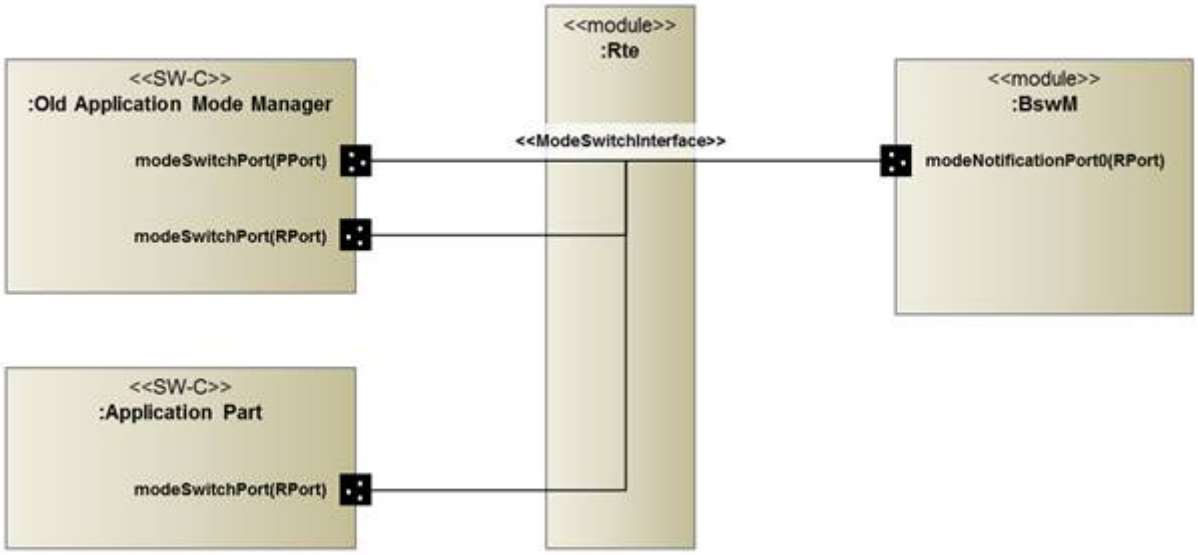


**Figure 7.5: Connections between SW-C based Application Mode Manager, Application Parts and the BSW Mode Manager**

Figure 7.5 shows that SW-C based Application Mode Managers (as used in AUTOSAR R3.1 and earlier) switch the application mode directly and do not request it from the BSW Mode Manager. Therefore, they directly connect a Mode Switch Port to the local RTE. This implies that the application mode needs to be local to that ECU and that no arbitration in the BSW Mode Manager is possible. Nevertheless, the BSW Mode Manager may use the current application mode as an input for its rules because it can have a Mode Switch R-Port (named `modeNotificationPort0` in the figure) for this application mode.

Note: To configure the BswM, knowledge of what mode request ports and ECU resources are needed/available for a specific ECU is needed. Therefore, the SW-C description of the BswM can only be completed during ECU configuration time.

From now on, all following interface definitions are interpreted to be in:

`ARPackage AUTOSAR_BswM/BswModuleDescription`

Note that the pseudocode presented in this chapter is not exact, but provides a hint of how the corresponding model elements need to be defined.

### 7.6.1 Mode Request Ports

The BSW Mode Manager must declare a Receiver Port with the interface defined in the context of the SW-C:

```
RequirePort AppModeRequestInterface modeRequest-
Port_{ArbName}_{ReqName};
```

To read the currently requested mode, the BSW Mode Manager implementation must call:

```
Rte_Read_modeRequestPort_{ArbName}_{ReqName}_requestedMode( &
<variable> );
```

### 7.6.2 Mode Switch Ports

As with Mode Requests, the BSW Mode Manager only references the mode switch interfaces defined in the context of the corresponding SW-C Description in its Provide Ports for mode switches. For the above example the declaration for a mode switch is:

```
ProvidePort AppModeInterface modeSwitchPort_{ModConName}
_{SwitchName};
```

The configuration parameter `BswMModeSwitchInterfaceRef` references this Mode Switch interface.

To switch the currently active mode, the BSW Mode Manager implementation must insert one of the following calls into its actions list:

```
Rte_Switch_modeSwitchPort_{ModConName}_{SwitchName}_currentMode
( <new_mode> );
```

```
SchM_Switch_modeSwitchPort_{ModConName}_{SwitchName}_current
Mode( <new_mode> );
```

### 7.6.3 Notifications of Mode Switches

In addition to mode requests, the currently active modes can also be used as inputs to mode arbitration. For Application and Vehicle Modes, the BSW Mode Manager needs to register as a mode user. It then receives notifications about changed modes via a Mode Switch Port. For the above example the declaration for a mode notification is:

Note: In order to make it easier to distinguish between a RequirePort and ProvidePort of the ModeSwitchPort type, the RequirePorts are named mode notification port for the following example.

```
RequirePort AppModeInterface modeNotificationPort_{Arb
Name}_{ModeName};
```

To read the currently active mode, the BSW Mode Manager implementation must call one of the following functions:

```
Rte_Mode_modeNotificationPort_{ArbName}_{ModeName}_currentMode(
&<variable> );
```

```
SchM_Mode_modeNotificationPort_{ArbName}_{ModeName}_currentMode
( &<variable> );
```

In case the enhanced Rte_Mode or SchM_Mode is configured, the BSW Mode Manager implementation must call one of the following functions:

```
Rte_Mode_modeNotificationPort_{ArbName}_{ModeName}_currentMode(
&<variable>, &<previousmode>, &<nextmode> );

SchM_Mode_modeNotificationPort_{ArbName}_{ModeName}_currentMode
( &<variable>, &<previosmode>, &<nextmode> );
```

### 7.6.4 Component Type and Internal Behavior

The BSW Mode Manager is a Service Component that serves Mode Requests local to the ECU. The ServiceComponentType for the BSW Mode Manager declares all of the above-mentioned ports, and some Internal Behavior.

```
ServiceComponentType BswM {
    ...
    InternalBehavior {
        ...
    };
};
```

The internal behavior depends on the parameter BswMRequestProcessing for the corresponding Mode Request Port. For BSWM_DEFERRED, the RTE must not perform any special actions, as the BSWM Mode Manager reads the request cyclically in its BswM_MainFunction. By contrast, for BSWM_IMMEDIATE the RTE must trigger mode arbitration immediately. Therefore, the BSW Mode Manager needs to register a trigger function that triggers mode arbitration. For the above example, an immediate processing of the mode request would need the following declaration in the Internal Behavior of the BSW Mode Manager:

```
RunnableEntity ModeArbitrationRunnable {
    symbol = <mode_arbitration_function>;
    canBeInvokedConcurrently = TRUE;
};


DataReceiveEvent AppModeRequestEvent {
    port = modeRequestPort0;
    dataElement = requestedMode;
    startOnEvent = ModeArbitrationRunnable;
};
```

Note: To deal with Mode Requests that originate from other ECUs, another kind of service component is needed. On the VFB level it looks like one global Service Component, but actually it is instantiated as one Service Component that resides above the RTE for each ECU. To support that, the SW-C Template offers the ServiceProxy ComponentType instead of the normal ServiceComponentType.

The specification of the Mode Management Service Proxy Component is not described within this document since it is user specific.

## 7.7 Ethernet switch port group switching

The current version of the SWS BswM supports Ethernet switch port group switching. Based on the current requested PNC, BswM maps the PNC request to a configured Eth IfSwitchPortGroup and calls `EthIf_SwitchPortGroupRequestMode`.The BswM is indicated by the EthIf, if the accumulated link state has changed. The accumulated link state could be used to inform an application. This could be used to cover error scenarios where the requested and the current accumulated link state of EthIfSwitch PortGroups are contradicting and error handling needs to be initiated.

The AUTOSAR_EXP_ModemanagementGuide document contains guidelines for the BswM configuration regarding Ethernet switch port group switching.

# 8 API specification

## 8.1 Imported types

**[SWS_BswM_00237]** ⌈The BSW Mode Manager shall use only the imported types which are listed in SWS_BswM_00001.⌋*(SRS_BSW_00301)*

**[SWS_BswM_00001]** ⌈

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| CanSM | CanSM.h | CanSM_BswMCurrentStateType |
| Com | Com.h | Com_IpduGroupIdType |
| ComM | ComM.h | ComM_InitStatusType |
| | ComM.h | ComM_PncModeType |
| | Rte_ComM_Type.h | ComM_InhibitionStatusType |
| | Rte_ComM_Type.h | ComM_ModeType |
| | Rte_ComM_Type.h | ComM_UserHandleType |
| ComStack_Types | ComStack_Types.h | NetworkHandleType |
| | ComStack_Types.h | PNCHandleType |
| | ComStack_Types.h | PduIdType |
| Dcm | Dcm.h | Dcm_CommunicationModeType |
| EcuM | EcuM.h | EcuM_RunStatusType |
| | EcuM.h | EcuM_StateType |
| | EcuM.h | EcuM_WakeupSourceType |
| | EcuM.h | EcuM_WakeupStatusType |
| | Rte_EcuM_Type.h | EcuM_UserType |

▽

△

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| Eth | Eth_GeneralTypes.h | Eth_ModeType (draft) |
| EthIf | EthIf.h | EthIf_SwitchPortGroupIdxType |
| EthSM | EthSM.h | EthSM_NetworkModeStateType |
| EthTrcv | Eth_GeneralTypes.h | EthTrcv_LinkStateType |
| FrSm | FrSm.h | FrSM_BswM_StateType |
| IdsM | IdsM_Filters_Types.h | IdsM_Filters_BlockStateType |
| J1939Dcm | J1939Dcm.h | J1939Dcm_StateType |
| J1939Rm | J1939Rm.h | J1939Rm_StateType |
| LinIf | LinIf.h | LinIf_SchHandleType |
| | LinIf.h | LinTp_Mode |
| LinSM | LinSM.h | LinSM_ModeType |
| McOs | Os.h | CoreIdType |
| Nm | NmStack_types.h | Nm_StateType |
| NvM | NvM.h | NvM_MultiBlockRequestType |
| | Rte_NvM_Type.h | NvM_BlockIdType |
| | Rte_NvM_Type.h | NvM_RequestResultType |
| Os | Os.h | ApplicationType |
| | Os.h | IdleModeType |
| | Os.h | StatusType |
| Sd | Sd.h | Sd_ClientServiceCurrentStateType |
| | Sd.h | Sd_ConsumedEventGroupCurrentStateType |
| | Sd.h | Sd_EventHandlerCurrentStateType |
| | Sd.h | Sd_ServiceGroupIdType |
| SoAd | SoAd.h | SoAd_SoConIdType |
| | SoAd.h | SoAd_SoConModeType |
| Std | Std_Types.h | Std_ReturnType |
| | Std_Types.h | Std_VersionInfoType |

⌋(*SRS_BSW_00301*)

## 8.2 Type definitions

**[SWS_BswM_00041]** ⌈The following Data Types shall be used for the functions defined in this specification.⌋(*SRS_BSW_00305*)

### 8.2.1 BswM_ConfigType

**[SWS_BswM_00213]** ⌈

| | | |
|---|---|---|
| ***Name*** | BswM_ConfigType | |
| ***Kind*** | Structure | |
| ***Elements*** | - | |
| | ***Type*** | – |
| | ***Comment*** | The contents of this structure depends on the configuration variant. |
| ***Description*** | This structure contains all post-build configurable parameters of the BSW Mode Manager. A pointer to this structure is passed to the BSW Mode Manager initialization function for configuration. | |
| ***Available via*** | BswM.h | |

⌋(*SRS_BSW_00404*, *SRS_BSW_00441*)

**[SWS_BswM_00042]** ⌈The structure `BswM_ConfigType` shall contain all post-build configurable parameters of the BSW Mode Manager. The exact content of this structure depends on the selected configuration variant⌋(*SRS_BSW_00404*)

### 8.2.2 BswM_ModeType

**[SWS_BswM_00214]** ⌈

| | | | |
|---|---|---|---|
| ***Name*** | BswM_ModeType | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint16 | | |
| ***Range*** | 0-65535 | – | – |
| ***Description*** | This type identifies the modes that can be requested by BswM Users. | | |
| ***Available via*** | BswM.h | | |

⌋(*SRS_ModeMgm_09228*, *SRS_BSW_00441*)

### 8.2.3 BswM_UserType

**[SWS_BswM_00216]** ⌈

| | | | |
|---|---|---|---|
| ***Name*** | BswM_UserType | | |
| ***Kind*** | Type | | |
| ***Derived from*** | uint16 | | |
| ***Range*** | 0-65535 | – | – |
| ***Description*** | This type identifies a BswM User that makes mode requests to the BswM. | | |
| ***Available via*** | BswM.h | | |

⌋(*SRS_ModeMgm_09228*, *SRS_BSW_00441*)

## 8.3 Function definitions

### 8.3.1 BswM_BswMPartitionRestarted

**[SWS_BswM_00193]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_BswMPartitionRestarted |
| ***Syntax*** | `void BswM_BswMPartitionRestarted (`<br>`  void`<br>`)` |
| ***Service ID [hex]*** | 0x1e |
| ***Sync/Async*** | Synchronous |
| ***Reentrancy*** | Reentrant |
| ***Parameters (in)*** | None |
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | Function called by Restart Task if the partition containing the BswM has been restarted. |
| ***Available via*** | BswM.h |

⌋(*SRS_ModeMgm_09228*)

### 8.3.2 BswM_CanSM_CurrentState

**[SWS_BswM_00049]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_CanSM_CurrentState | |
| ***Syntax*** | `void BswM_CanSM_CurrentState (`<br>`  NetworkHandleType Network,`<br>`  CanSM_BswMCurrentStateType CurrentState`<br>`)` | |
| ***Service ID [hex]*** | 0x05 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The CAN channel that the indicated state corresponds to. |
| | CurrentState | The current state of the CAN channel. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by CanSM to indicate its current state. | |
| ***Available via*** | BswM_CanSM.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00080]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall

ignore the state indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00095]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentState shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.3 BswM_ComM_CurrentMode

**[SWS_BswM_00047]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_ComM_CurrentMode | |
| ***Syntax*** | `void BswM_ComM_CurrentMode ( NetworkHandleType Network, ComM_ModeType RequestedMode )` | |
| ***Service ID [hex]*** | 0x0e | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The ComM communication channel that the indicated state corresponds to. |
| | RequestedMode | The current state of the ComM communication channel |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by ComM to indicate the current communication mode of a ComM channel. | |
| ***Available via*** | BswM_ComM.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00078]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00091]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter RequestedMode shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.4 BswM_ComM_CurrentPNCMode

**[SWS_BswM_00148]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_ComM_CurrentPNCMode | |
| ***Syntax*** | `void BswM_ComM_CurrentPNCMode ( PNCHandleType PNC, ComM_PncModeType CurrentPncMode )` | |
| ***Service ID [hex]*** | 0x15 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | PNC | The handle of the PNC for which the current state is reported. |
| | CurrentPncMode | The current mode of the PNC. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function is called by ComM to indicate the current mode of the PNC. | |
| ***Available via*** | BswM_ComM.h | |

⌋(*SRS_ModeMgm_09228*, *SRS_ModeMgm_09240*)

**[SWS_BswM_00149]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00150]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentPncMode shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.5 BswM_ComM_InitiateReset

**[SWS_BswM_00217]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_ComM_InitiateReset |
| ***Syntax*** | `void BswM_ComM_InitiateReset ( void )` |
| ***Service ID [hex]*** | 0x22 |
| ***Sync/Async*** | Synchronous |
| ***Reentrancy*** | Non Reentrant |
| ***Parameters (in)*** | None |
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | Function is called by ComM to signal a shutdown. |
| ***Available via*** | BswM_ComM.h |

⌋(*SRS_ModeMgm_09228*)

### 8.3.6 BswM_Dcm_ApplicationUpdated

**[SWS_BswM_00158]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_Dcm_ApplicationUpdated |
| ***Syntax*** | `void BswM_Dcm_ApplicationUpdated ( void )` |
| ***Service ID [hex]*** | 0x14 |
| ***Sync/Async*** | Synchronous |
| ***Reentrancy*** | Reentrant |
| ***Parameters (in)*** | None |
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | This function is called by the DCM in order to report an updated application. |
| ***Available via*** | BswM_Dcm.h |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00159]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

### 8.3.7 BswM_Dcm_CommunicationMode_CurrentState

**[SWS_BswM_00048]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Dcm_CommunicationMode_CurrentState | |
| ***Syntax*** | `void BswM_Dcm_CommunicationMode_CurrentState ( NetworkHandleType Network, Dcm_CommunicationModeType RequestedMode )` | |
| ***Service ID [hex]*** | 0x06 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The communication channel that the diagnostic mode corresponds to. |
| | RequestedMode | The requested diagnostic communication mode. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |

▽

△

| | |
|---|---|
| ***Return value*** | None |
| ***Description*** | Function called by DCM to inform the BswM about the current state of the communication mode. |
| ***Available via*** | BswM_Dcm.h |

⌋(*SRS_ModeMgm_09228*)

The behavior of this function shall be configured using the configuration container `BswMDcmComModeRequest`, wherein the configuration parameter `BswMDcmComM-ChannelRef` correlates to the argument Network of this function.

**[SWS_BswM_00079]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00093]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter RequestedMode shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

CDD Implementation Hint: All AUTOSAR BSW modules that may trigger transmission of PDUs provide an API to enable/disable it. To e.g. disable the whole communication in a corresponding diagnostic request, it makes sense if CDD modules (which use communication protocols) provides such an API as well. These functions may be called in the configured action list which is linked to this function.

### 8.3.8 BswM_Deinit

**[SWS_BswM_00119]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_Deinit |
| ***Syntax*** | `void BswM_Deinit ( void )` |
| ***Service ID [hex]*** | 0x04 |
| ***Sync/Async*** | Synchronous |
| ***Reentrancy*** | Non Reentrant |
| ***Parameters (in)*** | None |
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | Deinitializes the BSW Mode Manager. |
| ***Available via*** | BswM.h |

⌋(*SRS_BSW_00336*)

**[SWS_BswM_00120]** ⌈After a call of `BswM_Deinit` no mode processing shall be performed by BswM even if any mode requests are made or the BswM main function is called.⌋ *(SRS_BSW_00336)*

### 8.3.9 BswM_EcuM_CurrentState

**[SWS_BswM_91003]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_EcuM_CurrentState | |
| ***Syntax*** | `void BswM_EcuM_CurrentState ( EcuM_StateType CurrentState )` | |
| ***Service ID [hex]*** | 0x28 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | CurrentState | The requested ECU Operation Mode |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called EcuM to indicate the current ECU Operation Mode. | |
| ***Available via*** | BswM_EcuM.h | |

⌋ *(SRS_ModeMgm_09228)*

**[SWS_BswM_00084]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋ *(SRS_BSW_00406)*

**[SWS_BswM_00103]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentState shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋ *(SRS_BSW_00323)*

### 8.3.10 BswM_EcuM_CurrentWakeup

**[SWS_BswM_00131]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_EcuM_CurrentWakeup |
| ***Syntax*** | `void BswM_EcuM_CurrentWakeup ( EcuM_WakeupSourceType source, EcuM_WakeupStatusType state )` |
| ***Service ID [hex]*** | 0x10 |

▽

△

| | | |
|---|---|---|
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | source | Wakeup source(s) that changed state. |
| | state | The new state of the wakeup source(s) |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by EcuM to indicate the current state of a wakeup source. | |
| ***Available via*** | BswM_EcuM.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00132]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00133]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter state shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.11 BswM_EcuM_RequestedState

**[SWS_BswM_91004]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_EcuM_RequestedState | |
| ***Syntax*** | `void BswM_EcuM_RequestedState ( EcuM_StateType State, EcuM_RunStatusType CurrentState )` | |
| ***Service ID [hex]*** | 0x29 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | State | The requested state by EcuM |
| | CurrentState | Result of the Run Request Protocol |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by EcuM notify about current Status of the Run Request Protocol. | |
| ***Available via*** | BswM_EcuM.h | |

⌋*(SRS_ModeMgm_09116, SRS_ModeMgm_09228)*

**[SWS_BswM_00227]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall

ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00228]** ⌈If the BswMDevErrorDetect switch is enabled, the parameter State shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

**[SWS_BswM_00229]** ⌈If the BswMDevErrorDetect switch is enabled, the parameter CurrentStatus shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.12 BswM_EthIf_PortGroupLinkStateChg

**[SWS_BswM_91001]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_EthIf_PortGroupLinkStateChg | |
| ***Syntax*** | `void BswM_EthIf_PortGroupLinkStateChg ( EthIf_SwitchPortGroupIdxType PortGroupIdx, EthTrcv_LinkStateType PortGroupState )` | |
| ***Service ID [hex]*** | 0x26 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | PortGroupIdx | The port group index in the context of the Ethernet Interface |
| | PortGroupState | The state of the port group. State is derived from the physical link of the Ethernet Transceiver: ETHTRCV_LINK_STATE_DOWN == Port group has link down. ETHTRCV_LINK_STATE_ACTIVE == Port group has link up. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by EthIf to indicate the link state change of a certain Ethernet switch port group. | |
| ***Available via*** | BswM_EthIf.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00267]** ⌈If the BswMDevErrorDetect switch is enabled, the BswM_EthIf_PortGroupLinkStateChg routine shall check if the BSW Mode Manager is initialized. In case the BswM is not initialized, the BswM shall ignore the mode request and report to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00268]** ⌈If the BswMDevErrorDetect switch is enabled, the BswM_EthIf_PortGroupLinkStateChg parameters shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.13 BswM_EthSM_CurrentState

**[SWS_BswM_00050]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_EthSM_CurrentState | |
| ***Syntax*** | `void BswM_EthSM_CurrentState ( NetworkHandleType Network, EthSM_NetworkModeStateType CurrentState )` | |
| ***Service ID [hex]*** | 0x0d | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The Ethernet channel that the indicated state corresponds to. |
| | CurrentState | The current state of the Ethernet channel. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by EthSM to indicate its current state. | |
| ***Available via*** | BswM_EthSM.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00081]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00097]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentState shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.14 BswM_FrSM_CurrentState

**[SWS_BswM_00051]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_FrSM_CurrentState | |
| ***Syntax*** | `void BswM_FrSM_CurrentState ( NetworkHandleType Network, FrSM_BswM_StateType CurrentState )` | |
| ***Service ID [hex]*** | 0x0c | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The FlexRay cluster that the indicated state corresponds to. |
| | CurrentState | The corrent state of the FlexRay cluster. |

▽

△

| | |
|---|---|
| ***Parameters (inout)*** | None |
| ***Parameters (out)*** | None |
| ***Return value*** | None |
| ***Description*** | Function called by FrSM to indicate its current state. |
| ***Available via*** | BswM_FrSM.h |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00082]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00099]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentState shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.15 BswM_GetVersionInfo

**[SWS_BswM_00003]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_GetVersionInfo | |
| ***Syntax*** | `void BswM_GetVersionInfo ( Std_VersionInfoType* VersionInfo )` | |
| ***Service ID [hex]*** | 0x01 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | VersionInfo | Pointer to where to store the version information of the module. |
| ***Return value*** | None | |
| ***Description*** | Returns the version information of this module. | |
| ***Available via*** | BswM.h | |

⌋*(SRS_BSW_00407, SRS_BSW_00003)*

### 8.3.16 BswM_Init

**[SWS_BswM_00002]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Init | |
| ***Syntax*** | `void BswM_Init (`<br>`  const BswM_ConfigType * ConfigPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x00 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Conditionally Reentrant | |
| ***Parameters (in)*** | ConfigPtr | Pointer to post-build configuration data |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Initializes the BSW Mode Manager. | |
| ***Available via*** | BswM.h | |

⌋(*SRS_BSW_00344*, *SRS_BSW_00404*, *SRS_BSW_00405*, *SRS_BSW_00101*, *SRS_BSW_00358*, *SRS_BSW_00414*)

The Reentrancy for the `BswM_Init` API is set to "Conditionally reentrant" for support of implementations where the BswM is instantiated in multiple partitions (e.g. in single core systems using multiple BSW partitions OR in multi-core systems.

**[SWS_BswM_00043]** ⌈This routine initializes the BSW Mode Manager. After execution of this routine the BSW Mode Manager is ready to arbitrate incoming mode requests.⌋ *(SRS_BSW_00101)*

**[SWS_BswM_00044]** ⌈This routine shall initialize all module global variables of the BSW Mode Manager.⌋*(SRS_BSW_00101)*

**[SWS_BswM_00118]** ⌈BswM_Init shall only require the OS and the SchM to be initialized before it can be called.⌋*(SRS_BSW_00467)*

**[SWS_BswM_00045]** ⌈If the `BswMDevErrorDetect` switch is enabled, the contents of the given configuration set shall be checked for being within the allowed boundaries. If an error is detected the initialization of the BSW Mode Manager shall not be executed and the error shall be reported to the Default Error Tracer with the value BSWM_E_PARAM_CONFIG.⌋*(SRS_BSW_00323)*

### 8.3.17 BswM_J1939DcmBroadcastStatus

**[SWS_BswM_00165]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_J1939DcmBroadcastStatus |
| ***Syntax*** | `void BswM_J1939DcmBroadcastStatus (`<br>`  uint16 NetworkMask`<br>`)` |

▽

△

| | | |
|---|---|---|
| ***Service ID [hex]*** | 0x1b | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | NetworkMask | Mask containing one bit for each available network. The bit position within this mask corresponds to the ComMChannel.ComMChannelId for the communication channel (so ComMChannelID 0 is represented by bit 0). The meaning for each bit is: 1: Network enabled, 0: Network disabled. Note: only the first 16 communication channel IDs can be supported by this API. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This API tells the BswM the desired communication status of the available networks. The status will typically be activated via COM I-PDU group switches. | |
| ***Available via*** | BswM_J1939Dcm.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00249]** ⌈The `BswM_J1939DcmBroadcastStatus` parameter NetworkMask is a bitmask where the bit position corresponds to the `ComMChannel.ComMChannelId` which is referenced by the `BswMJ1939DcmBroadcastStatus.BswMJ1939DcmChannelRef` parameter. For rule processing, the BswM shall use the value in NetworkMask of the bit (0 or 1) which lies in the position configured by the referenced `ComMChannel.ComMChannelId`.⌋(*SRS_ModeMgm_09228*)

### 8.3.18 BswM_J1939Nm_StateChangeNotification

**[SWS_BswM_00194]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_J1939Nm_StateChangeNotification | |
| ***Syntax*** | `void BswM_J1939Nm_StateChangeNotification ( NetworkHandleType Network, uint8 Node, Nm_StateType NmState )` | |
| ***Service ID [hex]*** | 0x18 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | Identification of the J1939 channel |
| | Node | Identification of the J1939 node |
| | NmState | Current (new) state of the J1939 node |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Notification of current J1939Nm state after state changes. | |
| ***Available via*** | BswM_J1939Nm.h | |

⌋(*SRS_ModeMgm_09228*)

### 8.3.19 BswM_LinSM_CurrentSchedule

**[SWS_BswM_00058]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_LinSM_CurrentSchedule | |
| ***Syntax*** | `void BswM_LinSM_CurrentSchedule ( NetworkHandleType Network, LinIf_SchHandleType CurrentSchedule )` | |
| ***Service ID [hex]*** | 0x0a | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The LIN channel that the schedule table switch have occurred on. |
| | CurrentSchedule | The currently active schedule table of the LIN channel. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by LinSM to indicate the currently active schedule table for a specific LIN channel. | |
| ***Available via*** | BswM_LinSM.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00086]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error the BswM shall ignore the schedule indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

### 8.3.20 BswM_LinSM_CurrentState

**[SWS_BswM_00052]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_LinSM_CurrentState | |
| ***Syntax*** | `void BswM_LinSM_CurrentState ( NetworkHandleType Network, LinSM_ModeType CurrentState )` | |
| ***Service ID [hex]*** | 0x09 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The LIN channel that the indicated state corresponds to. |
| | CurrentState | The current state of the LIN channel. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by LinSM to indicate its current state. | |

▽

△

| | |
|---|---|
| ***Available via*** | BswM_LinSM.h |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00083]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00101]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentState shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the state indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.21 BswM_LinTp_RequestMode

**[SWS_BswM_00156]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_LinTp_RequestMode | |
| ***Syntax*** | `void BswM_LinTp_RequestMode ( NetworkHandleType Network, LinTp_Mode LinTpRequestedMode )` | |
| ***Service ID [hex]*** | 0x0b | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | The LIN channel that the LinTp mode request relates to. |
| | LinTpRequestedMode | The requested LIN TP mode. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by LinTP to request a mode for the corresponding LIN channel. The LinTp_Mode correlates to the LIN schedule table that should be used. | |
| ***Available via*** | BswM_LinTp.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00112]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00113]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter LinTpRequestedMode shall be checked for being in the allowed range. In case of an error the BswM shall ignore the mode request and report the error, to the Default

Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_-00323*)

### 8.3.22 BswM_Nm_CarWakeUpIndication

**[SWS_BswM_00235]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Nm_CarWakeUpIndication | |
| ***Syntax*** | `void BswM_Nm_CarWakeUpIndication ( NetworkHandleType Network )` | |
| ***Service ID [hex]*** | 0x24 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | Identification of the Nm-Channel |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by Nm to indicate a CarWakeup. | |
| ***Available via*** | BswM_Nm.h | |

⌋(*SRS_ModeMgm_09228*)

### 8.3.23 BswM_Nm_StateChangeNotification

**[SWS_BswM_91002]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Nm_StateChangeNotification | |
| ***Syntax*** | `void BswM_Nm_StateChangeNotification ( NetworkHandleType Network, Nm_StateType currentState )` | |
| ***Service ID [hex]*** | 0x27 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Network | Identification of the Nm-channel |
| | currentState | Current (new) state of the Nm-channel |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Notification of current Nm state after state changes. | |
| ***Available via*** | BswM_Nm.h | |

⌋()

### 8.3.24 BswM_NvM_CurrentBlockMode

**[SWS_BswM_00104]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_NvM_CurrentBlockMode | |
| ***Syntax*** | `void BswM_NvM_CurrentBlockMode ( NvM_BlockIdType Block, NvM_RequestResultType CurrentBlockMode )` | |
| ***Service ID [hex]*** | 0x16 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Block | The Block that the new NvM Mode corresponds to. |
| | CurrentBlockMode | The current block mode of the NvM block. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by NvM to indicate the current block mode of an NvM block. | |
| ***Available via*** | BswM_NvM.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00109]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the block mode indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00110]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentBlockMode shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the block mode indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.25 BswM_NvM_CurrentJobMode

**[SWS_BswM_00152]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_NvM_CurrentJobMode | |
| ***Syntax*** | `void BswM_NvM_CurrentJobMode ( NvM_MultiBlockRequestType MultiBlockRequest, NvM_RequestResultType CurrentJobMode )` | |
| ***Service ID [hex]*** | 0x17 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | MultiBlockRequest | Indicates which multi block service this callback refers to. |

▽

△

| | | |
|---|---|---|
| | CurrentJobMode | Current state of the multi block job indicated by parameter Multi BlockRequest |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by NvM to inform the BswM about the current state of a multi block job. | |
| ***Available via*** | BswM_NvM.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00153]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the job mode indication and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00154]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter MultiBlockRequest shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the job mode indication and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.26 BswM_RequestMode

**[SWS_BswM_00046]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_RequestMode | |
| ***Syntax*** | `void BswM_RequestMode ( BswM_UserType requesting_user, BswM_ModeType requested_mode )` | |
| ***Service ID [hex]*** | 0x02 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | requesting_user | The user that requests the mode |
| | requested_mode | The requested mode. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Generic function call to request modes. This function shall only be used by other BSW modules that does not have a specific mode request interface. | |
| ***Available via*** | BswM.h | |

⌋(*SRS_ModeMgm_09179, SRS_ModeMgm_09228*)

**[SWS_BswM_00077]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall

ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00089]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter requested_mode shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

**[SWS_BswM_00090]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter requesting_user shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.27 BswM_Sd_ClientServiceCurrentState

**[SWS_BswM_00204]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Sd_ClientServiceCurrentState | |
| ***Syntax*** | `void BswM_Sd_ClientServiceCurrentState ( uint16 SdClientServiceHandleId, Sd_ClientServiceCurrentStateType CurrentClientState )` | |
| ***Service ID [hex]*** | 0x1f | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | SdClientServiceHandleId | HandleId to identify the ClientService |
| | CurrentClientState | Current state of the ClientService |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by Service Discovery to indicate current state of the Client Service (available/ down). | |
| ***Available via*** | BswM_Sd.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00205]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00206]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter CurrentClientState shall be checked for being in the allowed range. In case of an error the BswM shall ignore the mode request and report the error, to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.28 BswM_Sd_ConsumedEventGroupCurrentState

**[SWS_BswM_00207]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_Sd_ConsumedEventGroupCurrentState | |
| ***Syntax*** | `void BswM_Sd_ConsumedEventGroupCurrentState ( uint16 SdConsumedEventGroupHandleId, Sd_ConsumedEventGroupCurrentStateType ConsumedEventGroupState )` | |
| ***Service ID [hex]*** | 0x21 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | SdConsumedEventGroup HandleId | HandleId to identify the Consumed Eventgroup |
| | ConsumedEventGroup State | Status of the Consumed Eventgroup |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by Service Discovery to indicate current status of the Consumed Eventgroup (available/down). | |
| ***Available via*** | BswM_Sd.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00208]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

**[SWS_BswM_00209]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter ConsumedEventGroupState shall be checked for being in the allowed range. In case of an error the BswM shall ignore the mode request and report the error, to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋*(SRS_BSW_00323)*

### 8.3.29 BswM_Sd_EventHandlerCurrentState

**[SWS_BswM_00210]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_Sd_EventHandlerCurrentState |
| ***Syntax*** | `void BswM_Sd_EventHandlerCurrentState ( uint16 SdEventHandlerHandleId, Sd_EventHandlerCurrentStateType EventHandlerStatus )` |
| ***Service ID [hex]*** | 0x20 |
| ***Sync/Async*** | Synchronous |

▽

△

| | | |
|---|---|---|
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | SdEventHandlerHandleId | HandleId to identify the EventHandler |
| | EventHandlerStatus | Status of the EventHandler |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by Service Discovery to indicate current status of the EventHandler (requested/ released). | |
| ***Available via*** | BswM_Sd.h | |

⌋(*SRS_ModeMgm_09228*)

**[SWS_BswM_00211]** ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋(*SRS_BSW_00406*)

**[SWS_BswM_00212]** ⌈If the `BswMDevErrorDetect` switch is enabled, the parameter EventHandlerStatus shall be checked for being in the allowed range. In case of an error the BswM shall ignore the mode request and report the error, to the Default Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋(*SRS_BSW_00323*)

### 8.3.30 BswM_SoAd_SoConModeChg

**[SWS_BswM_91005]** ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_SoAd_SoConModeChg | |
| ***Syntax*** | `void BswM_SoAd_SoConModeChg (`<br>`  SoAd_SoConIdType SoConId,`<br>`  SoAd_SoConModeType State`<br>`)` | |
| ***Service ID [hex]*** | 0x2a | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different SoConIds. Non reentrant for the same SoConId. | |
| ***Parameters (in)*** | SoConId | The socket connection index. |
| | State | The state of the SoAd socket connection. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by SoAd to notify state changes of a socket connection. | |
| ***Available via*** | BswM_SoAd.h | |

⌋()

### 8.3.31 BswM_WdgM_RequestPartitionReset

**[SWS_BswM_00157]**{OBSOLETE} ⌈

| | | |
|---|---|---|
| ***Service Name*** | BswM_WdgM_RequestPartitionReset (obsolete) | |
| ***Syntax*** | `void BswM_WdgM_RequestPartitionReset ( ApplicationType Application )` | |
| ***Service ID [hex]*** | 0x11 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | Application | The identifier of an OS-Application |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Function called by WdgM to request a partition reset.<br>**Tags:** atp.Status=obsolete | |
| ***Available via*** | BswM_WdgM.h | |

⌋*(SRS_ModeMgm_09228)*

**[SWS_BswM_00134]**{OBSOLETE} ⌈If the `BswMDevErrorDetect` switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT.⌋*(SRS_BSW_00406)*

## 8.4 Call-back notifications

There are no call-back notifications in the BswM.

## 8.5 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non-reentrant.

### 8.5.1 BswM_MainFunction

**[SWS_BswM_00053]** ⌈

| | |
|---|---|
| ***Service Name*** | BswM_MainFunction |

▽

△

| | |
|---|---|
| ***Syntax*** | `void BswM_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x03 |
| ***Description*** | Main function of the BswM |
| ***Available via*** | SchM_BswM.h |

⌋(*SRS_BSW_00425*)

**[SWS_BswM_00075]** ⌈The `BswM_MainFunction` shall perform evaluation of all rules that uses at least one mode request with configuration parameter `BswMRequestProcessing` set to BSWM_DEFERRED as input.⌋(*SRS_ModeMgm_09180*)

## 8.6 Expected Interfaces

In this chapter all external interfaces required from other modules are listed.

### 8.6.1 Mandatory Interfaces

This chapter defines all interfaces that are required to fulfill the core functionality of the module.

**[SWS_BswM_00007]** ⌈

| ***API Function*** | ***Header File*** | ***Description*** |
|---|---|---|
| There are no mandatory interfaces. | | |

⌋(*SRS_BSW_00384*)

### 8.6.2 Optional Interfaces

According to [SWS_BswM_00039], the BswM can call any function in the AUTOSAR BSW. The following table contains a list of specific functions which may be useful in implementing BswM functionality.

**[SWS_BswM_00008]** ⌈

| ***API Function*** | ***Header File*** | ***Description*** |
|---|---|---|
| Com_DisableReceptionDM | Com.h | Disables the reception deadline monitoring for the I-PDUs within the given I-PDU group. |

▽

△

| API Function | Header File | Description |
|---|---|---|
| Com_EnableReceptionDM | Com.h | Enables the reception deadline monitoring for the I-PDUs within the given I-PDU group. |
| Com_IpduGroupStart | Com.h | Starts a preconfigured I-PDU group. For example, cyclic I-PDUs will be sent out cyclically after the call of Com_IpduGroupStart(). If Initialize is true all I-PDUs of the I-PDU group shall be (re-)initialized before the I-PDU group is started. That is they shall behave like after a start-up of COM, for example the old_value of the filter objects and shadow buffers of signal groups have to be (re-)initialized. |
| Com_IpduGroupStop | Com.h | Stops a preconfigured I-PDU group. For example, cyclic I-PDUs will be stopped after the call of Com_IpduGroupStop(). |
| Com_SwitchIpduTxMode | Com.h | The service Com_SwitchIpduTxMode sets the transmission mode of the I-PDU referenced by Pdu Id to Mode. In case the transmission mode changes, the new mode shall immediately be effective (see SWS_Com_00239). In case the requested transmission mode was already active for this I-PDU, the call will have no effect. |
| ComM_GetCurrentComMode | ComM.h | Function to query the current Communication Mode. ComM shall use the corresponding interfaces of the Bus State Managers to get the current Communication Mode of the network. (Call to Bus State Manager API: XXXSM _GetCurrentCom Mode(...)) |
| ComM_GetInhibitionStatus | ComM.h | Returns the inhibition status of a ComM channel. |
| ComM_GetMaxComMode | ComM.h | Function to query the maximum allowed Communication Mode of the corresponding user. |
| ComM_GetRequestedComMode | ComM.h | Function to query the currently requested Communication Mode of the corresponding user. |
| ComM_GetStatus | ComM.h | Returns the initialization status of the AUTOSAR Communication Manager. After a call to ComM_De Init() ComM should have status COMM_UNINIT, and a new call to ComM_Init needed to make sure Com M restart internal state machines to defailt values. |
| ComM_GetVersionInfo | ComM.h | This function returns the version information of this module |
| ComM_LimitChannelToNoComMode | ComM.h | Changes the inhibition status for the channel for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_ LimitECUToNoComMode, same functionality but for all channels) |
| ComM_LimitECUToNoComMode | ComM.h | Changes the inhibition status for the ECU (=all channels) for changing from COMM_NO_ COMMUNICATION to a higher Communication Mode. (See also ComM_LimitChannelToNoCom Mode, same functionality but for a specific channels) |
| ComM_PreventWakeUp | ComM.h | Changes the inhibition status COMM_NO_WAKEUP for the corresponding channel. |
| ComM_ReadInhibitCounter | ComM.h | This function returns the amount of rejected COMM_FULL_COMMUNICATION user requests. |
| ComM_RequestComMode | ComM.h | Requesting of a Communication Mode by a user.<br>Note:<br>The following modes are no vaild user requests, since they are used as internal modes:<br>▽ |

▽

△

| API Function | Header File | Description |
|---|---|---|
| | | △<br>- COMM_SILENT_COMMUNICATION (this mode is used for synchronization at shutdown)<br>- COMM_FULL_COMMUNICATION_WITH_ WAKEUP_REQUEST (this mode is used internally within the ComM channel statemachine to trigger the lower layers to request a wakeup on the network if the used hardware support such a feature. (e.g. Ethernet hardware which is compatible with OA TC10).<br>The following modes are valid user requests:<br>- COMM_NO_COMMUNICATION<br>- COMM_FULL_COMMUNICATION. The communication request could also be released due to a ComM communication inhibition |
| ComM_ResetInhibitCounter | ComM.h | This function resets the Inhibited COMM_FULL_ COMMUNICATION request Counter. |
| ComM_SetECUGroupClassification | ComM.h | Changes the ECU Group Classification status (see chapter 10.2.2) |
| ControlIdle | Os.h | This API allows the caller to select the idle mode action which is performed during idle time of the OS (e.g. if no Task/ISR is active). It can be used to implement energy savings. The real idle modes are hardware dependent and not standardized. The default idle mode on each core is IDLE_NO_HALT. |
| Det_ReportError | Det.h | Service to report development errors. |
| EcuM_AL_DriverInitBswM_<x> | EcuM.h | This callback shall provide BSW module initializations to be called by the BSW Mode Manager. |
| EcuM_GoDownHaltPoll | EcuM.h | Instructs the ECU State Manager module to go into a sleep mode, Reset or OFF depending on the previously selected shutdown target. |
| EcuM_SetState | EcuM.h | Function called by BswM to notify about State Switch. |
| EthIf_StartAllPorts | EthIf.h | Request to set all configured and affected EthSwt Ports to ETH_MODE_ACTIVE |
| EthIf_SwitchPortGroupRequestMode | EthIf.h | Request a mode for the EthIfSwtPortGroup. The call shall be forwarded to EthSwt by calling EthSwt_Set SwitchPortMode for all EthSwtPorts referenced by the port group. |
| FrSm_AllSlots | FrSm.h | This API function can be used to leave the KeySlot OnlyMode. |
| FrSm_SetEcuPassive | FrSm.h | This API function can be used to set all FlexRay clusters of the ECU to a receive only mode. |
| IdsM_BswM_StateChanged | IdsM_Cbk.h | This callback function is invoked by the BswM to indicate ECU state changes. |
| J1939Dcm_SetState | J1939Dcm.h | Changes the communication state of J1939Dcm to offline or online. |
| J1939Rm_SetState | J1939Rm.h | Changes the communication state of J1939Rm to offline (only Request for AC supported) or online. |
| LinSM_ScheduleRequest | LinSM.h | The upper layer requests a schedule table to be changed on one LIN network. |

▽

△

| API Function | Header File | Description |
|---|---|---|
| Nm_DisableCommunication | Nm.h | Disables the NM PDU transmission ability. For that purpose <BusNm>_DisableCommunication shall be called in case NmBusType is not set to NM_ BUSNM_LOCALNM (e.g. CanNm_Disable Communication function is called if channel is configured as CAN). |
| Nm_EnableCommunication | Nm.h | Enables the NM PDU transmission ability. For that purpose <BusNm>_EnableCommunication shall be called in case NmBusType is not set to NM_ BUSNM_LOCALNM. (e.g. CanNm_Enable Communication function is called if channel is configured as CAN). |
| Sd_ServiceGroupStart | Sd.h | Starts a preconfigured SdServiceGroup. For example, OfferService entries will be sent out after the call of Sd_ServiceGroupStart() for all Server Servives of a SdServiceGroup, which are not requested yet. |
| Sd_ServiceGroupStop | Sd.h | Stops a preconfigured SdServiceGroup. For example, StopOfferService entries will be sent out after the call of Sd_ServiceGroupStop() for all ServerServices of a SdServiceGroup, which are not requested by another SdServiceGroup. |

⌋(*SRS_BSW_00384*)

## 8.7 Service Interfaces

### 8.7.1 Scope of this Chapter

This chapter defines the AUTOSAR Interfaces of the Basic Software Mode Manager Service (BswM). The definitions in this section are interpreted to be in ARPackage AUTOSAR/Services/BswM.

### 8.7.2 Ports

#### 8.7.2.1 BswM_modeNotificationPort

**[SWS_BswM_00200]** ⌈

| Name | modeNotificationPort_{ArbName}_{ModeName} |
|---|---|
| Kind | RequiredPort |
| Interface-Ref | {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMModeRequestSource/ BswMSwcModeNotification.BswMSwcModeNotificationModeDeclarationGroupPrototype Ref)}.parent |
| Description | – |

▽

△

| | |
|---|---|
| ***Variation*** | ArbName = {ecuc(BswM/BswMConfig/BswMArbitration.SHORT-NAME)}<br>ModeName = {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMMode RequestSource/BswMSwcModeNotification.SHORT-NAME)} |

⌋(*SRS_ModeMgm_09180*)

**[SWS_BswM_00266]** ⌈If the `BswMDevErrorDetect` switch is enabled, BswM_modeNotificationPort shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the notification and report the error to the Default Error Tracer with the error code BSWM_E_UNINIT⌋(*SRS_BSW_00406*)

#### 8.7.2.2 BswM_modeRequestPort

**[SWS_BswM_00201]** ⌈

| | |
|---|---|
| ***Name*** | modeRequestPort_{ArbName}_{ReqName} |
| ***Kind*** | RequiredPort |
| ***Interface-Ref*** | {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort.BswMModeRequest Source.BswMSwcModeRequest.BswMSwcModeRequestVariableDataPrototypeRef)}.parent |
| ***Description*** | – |
| ***Variation*** | ArbName = {ecuc(BswM/BswMConfig/BswMArbitration.SHORT-NAME)}<br>ReqName = {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort.SHORT-NAME)} |

⌋(*SRS_ModeMgm_09179*)

#### 8.7.2.3 BswM_modeSwitchPort

**[SWS_BswM_00202]** ⌈

| | |
|---|---|
| ***Name*** | modeSwitchPort_{ModConName}_{SwitchName} |
| ***Kind*** | ProvidedPort |
| ***Interface-Ref*** | {ecuc(BswM/BswMConfig/BswMModeControl/BswMSwitchPort.BswMModeSwitchInterfaceRef)} |
| ***Description*** | – |
| ***Variation*** | {ecuc(BswM/BswMConfig/BswMModeControl/BswMSwitchPort.BswMModeSwitchInterfaceRef} != NULL<br>ModConName = {ecuc(BswM/BswMConfig/BswMModeControl.SHORT-NAME)}<br>SwitchName = {ecuc(BswM/BswMConfig/BswMModeControl/BswMSwitchPort.SHORT-NAME)} |

⌋(*SRS_ModeMgm_09182*)

## 8.8 API to Request Port Mappings

**[SWS_BswM_00283]** ⌈The following table describes which request port shall be triggered when an API is called. In addition, it describes which configuration parameter(s) correspond to which API parameter(s)

| API | Request Port | API / Config-parameter pairs |
|---|---|---|
| BswM_BswMPartitionRestarted | BswMPartitionRestarted | - |
| BswM_CanSM_CurrentState | BswMCanSMIndication | Network / BswMCanSMChannelRef |
| BswM_ComM_CurrentMode | BswMComMIndication | Network / BswMComMChannelRef |
| BswM_ComM_CurrentPNCMode | BswMComMPncRequest | PNC / BswMComMPncRef |
| BswM_ComM_InitiateReset | BswMComMInitiateReset | - |
| BswM_Dcm_ApplicationUpdated | BswMDcmApplicationUpdatedIndication | - |
| BswM_Dcm_CommunicationMode_CurrentState | BswMDcmComModeRequest | Network / BswMDcmComMChannelRef |
| BswM_EcuM_CurrentState | BswMEcuMIndication | - |
| BswM_EcuM_CurrentWakeup | BswMEcuMWakeupSource | source / BswMEcuMWakeupSrcRef |
| BswM_EcuM_RequestedState | BswMEcuMRUNRequestIndication | State / BswMEcuMRUNRequestProtocolPort |
| BswM_EthIf_PortGroupLinkStateChg | BswMEthIfPortGroupLinkStateChg | PortGroupIdx / BswMEthIfSwitchPortGroupRef |
| BswM_EthSM_CurrentState | BswMEthSMIndication | Network / BswMEthSMChannelRef |
| BswM_FrSM_CurrentState | BswMFrSMIndication | Network / BswMFrSMChannelRef |
| BswM_J1939DcmBroadcastStatus | BswMJ1939DcmBroadcastStatus | NetworkMask / BswMJ1939DcmChannelRef |
| BswM_J1939Nm_StateChangeNotification | BswMJ1939NmIndication | Network / BswMJ1939NmChannelRef, Node / BswMJ1939NmNodeRef |
| BswM_LinSM_CurrentSchedule | BswMLinScheduleIndication | Network / BswMLinSMChannelRef |
| BswM_LinSM_CurrentState | BswMLinSMIndication | Network / BswMLinSMChannelRef |
| BswM_LinTp_RequestMode | BswMLinTpModeRequest | Network / BswMLinTpChannelRef |
| BswM_Nm_CarWakeUpIndication | BswMNmCarWakeUpIndication | - |
| BswM_Nm_StateChangeNotification | BswMNmStateChangeNotification | Network / BswMNmChannelRef |
| BswM_NvM_CurrentBlockMode | BswMNvMRequest | Block / BswMNvMBlockRef |
| BswM_NvM_CurrentJobMode | BswMNvMJobModeIndication | MultiBlockRequest / BswMNvmService |
| BswM_RequestMode | BswMGenericRequest | requesting_user / BswMModeRequesterId |
| BswM_Sd_ClientServiceCurrentState | BswMSdClientServiceCurrentState | SdClientServiceHandleId / BswMSdClientMethodsRef |
| BswM_Sd_ConsumedEventGroupCurrentState | BswMSdConsumedEventGroupCurrentState | SdConsumedEventGroupHandleId / BswMSdConsumedEventGroupRef |

| API | Request Port | API / Config-parameter pairs |
|---|---|---|
| BswM_Sd_EventHandlerCurrentState | BswMSdEventHandlerCurrentState | SdEventHandlerHandleId / BswMSdEventHandlerRef |
| BswM_WdgM_RequestPartitionReset | BswMWdgMRequestPartitionReset | - |

**Table 8.1: API to Request Port Mappings**

⌋(*SRS_ModeMgm_09228*)

# 9 Sequence diagrams

## 9.1 Deferred operation of BswM

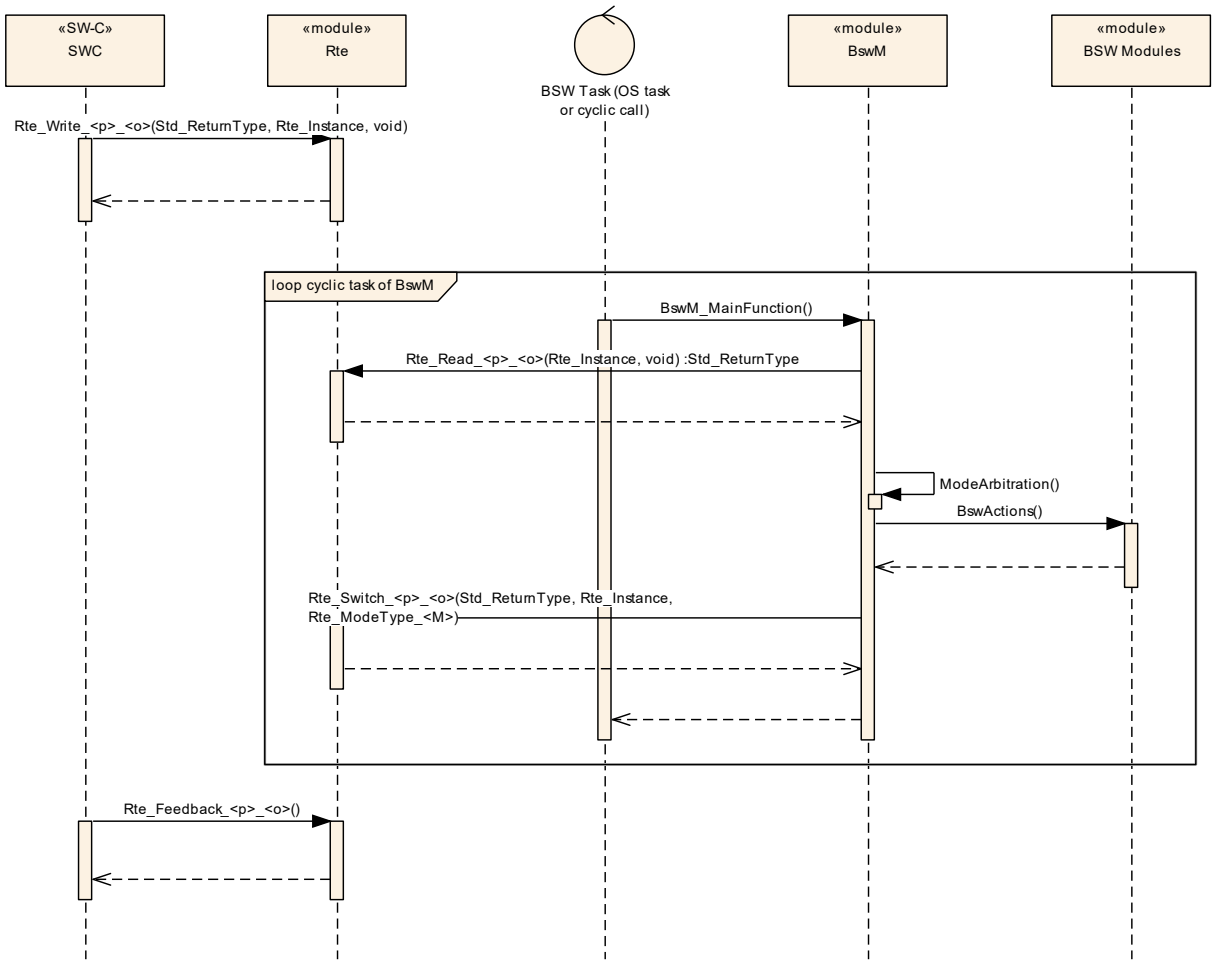


**Figure 9.1: Deferred operation of BswM**

## 9.2 Immediate operation of BswM

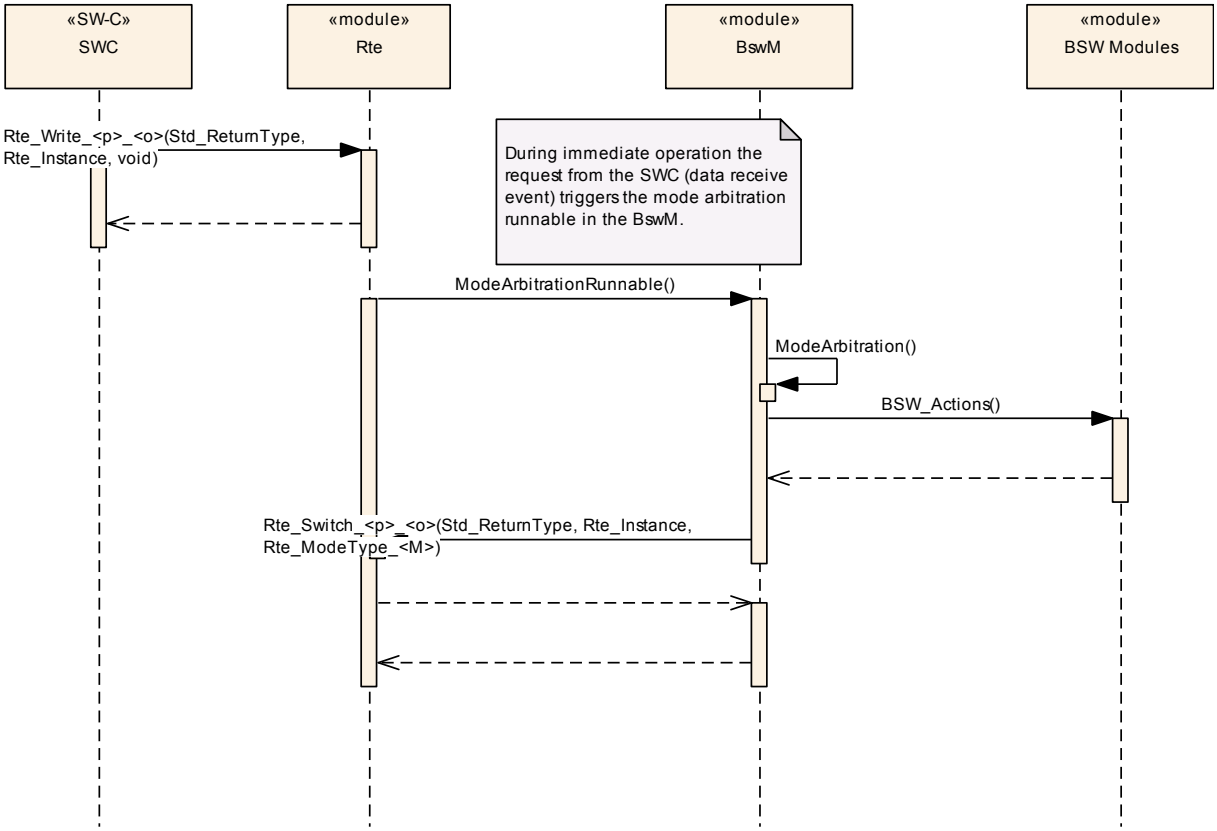


**Figure 9.2: Immediate operation of BswM**

# 10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module BSW Mode Manager.

Chapter 10.3 specifies published information of the module BSW Mode Manager.

## 10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in SWS_BSWGeneral.

## 10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapter 7 and Chapter 8.

Note for implementers: For some BswMModeRequestSources and BswMAvailableActions, a naming convention is used to map certain configuration parameter enumeration values with the underlying function parameter value used in the implementation. In this naming convention, the configuration parameter enumeration label is the same as the mapped function parameter enumeration label, but prefixed with "BSWM_". For example: the BswMEthIfSwitchPortGroupRequestMode value BSWM_ETH_MODE_ACTIVE corresponds to the value of ETH_MODE_ACTIVE of the respective type.

### 10.2.1 BswM

| **Module SWS Item** | ECUC_BswM_01063 | |
|---|---|---|
| **Module Name** | BswM | |
| **Module Description** | Configuration of the BswM (Basic SW Mode Manager) module. | |
| **Post-Build Variant Support** | true | |
| **Supported Config Variants** | VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE | |
| **Included Containers** | | |
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMConfig | 1..* | This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition. |
| BswMGeneral | 1 | General configuration parameters of the Basic SW Mode Manager. |

### 10.2.2 BswMConfig

| **SWS Item** | [ECUC_BswM_00895] |
|---|---|
| **Container Name** | BswMConfig |
| **Parent Container** | BswM |
| **Description** | This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition. |
| **Configuration Parameters** | |

| **Name** | BswMPartitionRef [ECUC_BswM_00984] |
|---|---|
| **Parent Container** | BswMConfig |
| **Description** | This references the partition the BswM shall run inside. |
| **Multiplicity** | 0..1 |
| **Type** | Reference to EcucPartition |
| **Post-Build Variant Multiplicity** | false |

| | | | |
|---|---|---|---|
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMArbitration | 1 | This container includes all configuration sub-containers and parameters related to the mode arbitration functionality of the BswM. |
| BswMDataTypeMappingSets | 0..1 | Collection of references to DataTypeMappingSet. |
| BswMModeControl | 1 | This container includes all configuration sub-containers and parameters related to the mode control functionality of the BswM. |

### 10.2.3 BswMArbitration

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00801] |
| **Container Name** | BswMArbitration |
| **Parent Container** | BswMConfig |
| **Description** | This container includes all configuration sub-containers and parameters related to the mode arbitration functionality of the BswM. |
| **Configuration Parameters** | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMEventRequestPort | 0..* | Each instance of this container defines an event which can be sent to the BswM. Basic Software Modules may send these events to the BswM by calling the corresponding BswM C-API (for example: BswM_ComM_InitiateReset()). |
| BswMLogicalExpression | 0..* | This container describes the logical expressions that can be used for the mode arbitration. The logical expressions are built of a set of arguments and a logical operator. Each argument can either be a mode condition or a sub-expression to allow definition of more complex logical expressions. There may be an unlimited number of arguments in each logical expression. Note that the order of evaluation of the expressions is not defined. |
| BswMModeCondition | 0..* | This container describes the BswM mode conditions that can be used either by itself to form a rule or as a part of a logical expression. |

| | | |
|---|---|---|
| BswMModeRequestPort | 0..* | Each instance of this container defines a mode request interface that is used to requests or indicate modes from/to the BswM. These interfaces are implemented as ports or as ordinary C-functions based upon if the request is made by an SW-C or a BSW module. There are different types of mode requests: 1. Mode requests from the SW-C:s 2. Mode Requests from other BSW modules such as the DCM. 3. State/mode indications from the RTE or other BSW modules such as the bus specific State Managers.<br><br>Note that the BswM treats all request and indications in the exact same way. |
| BswMRule | 0..* | Each instance of this container describes a BswM arbitration rule. The rule either consists of a simple mode condition or a more complex logical expression. This container also references the action lists that shall be invoked when the rule is evaluated to True or False. |

### 10.2.4 BswMLogicalExpression

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00808] | | |
| **Container Name** | BswMLogicalExpression | | |
| **Parent Container** | BswMArbitration | | |
| **Description** | This container describes the logical expressions that can be used for the mode arbitration. The logical expressions are built of a set of arguments and a logical operator. Each argument can either be a mode condition or a sub-expression to allow definition of more complex logical expressions. There may be an unlimited number of arguments in each logical expression. Note that the order of evaluation of the expressions is not defined. | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Configuration Parameters** | | | |

| | | |
|---|---|---|
| **Name** | BswMLogicalOperator [ECUC_BswM_00814] | |
| **Parent Container** | BswMLogicalExpression | |
| **Description** | This parameter specifies the logical operator to be used in the logical expression. If the logical operator is set to something other than BSWM_NOT, and the expression only consists of a single condition, then this parameter will have no effect. | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BSWM_AND | |
| | BSWM_NAND | |

| | | | |
|---|---|---|---|
| | BSWM_NOT | | |
| | BSWM_OR | | |
| | BSWM_XOR | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMArgumentRef [ECUC_BswM_00820] | | |
| **Parent Container** | BswMLogicalExpression | | |
| **Description** | This is a choice reference either to a mode condition or a sub-expression.<br><br>In case the BswMLogicalExpression.BswMLogicalOperator equals BSWM_NAND only two operands are supported. In case the BswMLogicalExpression.BswMLogicalOperator equals BSWM_NOT only one operand is supported. | | |
| **Multiplicity** | 1..* | | |
| **Type** | Choice reference to [BswMLogicalExpression, BswMModeCondition] | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

**[SWS_BswM_00242]** ⌈The BswM shall reject configurations where a `BswMLogicalExpression` has a `BswMLogicalOperator` equal to BSWM_NAND and its number of `BswMArgumentRef`s is not two.⌋ *(SRS_BSW_00167)*

**[SWS_BswM_00243]** ⌈The BswM shall reject configurations where a BswMLogical-Expression has a BswMLogicalOperator equal to BSWM_NOT and its number of BswMArgumentRefs is not one.⌋*(SRS_BSW_00167)*

**[SWS_BswM_00244]** ⌈The BswM shall implement BSWM_XOR to evaluate to TRUE if an odd number of its arguments is TRUE, and evaluate to FALSE if an even number of its arguments is TRUE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00245]** ⌈The BswM shall implement BSWM_AND to evaluate to TRUE if all of its arguments are TRUE, and evaluate to FALSE if at least one of its arguments is FALSE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00246]** ⌈The BswM shall implement BSWM_NAND to evaluate to FALSE if all of its arguments are TRUE, and evaluate to TRUE if at least one of its arguments is FALSE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00247]** ⌈The BswM shall implement BSWM_OR to evaluate to FALSE if all of its arguments are FALSE, and evaluate to TRUE if at least one of its arguments is TRUE.⌋*(SRS_ModeMgm_09180)*

**[SWS_BswM_00248]** ⌈The BswM shall implement BSWM_NOT to evaluate to FALSE if its argument is TRUE, and evaluate to TRUE if its argument is FALSE.⌋*(SRS_-ModeMgm_09180)*

### 10.2.5 BswMModeCondition

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00807] | | |
| **Container Name** | BswMModeCondition | | |
| **Parent Container** | BswMArbitration | | |
| **Description** | This container describes the BswM mode conditions that can be used either by itself to form a rule or as a part of a logical expression. | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Configuration Parameters** | | | |

| | | | |
|---|---|---|---|
| **Name** | BswMConditionType [ECUC_BswM_00815] | | |
| **Parent Container** | BswMModeCondition | | |
| **Description** | This parameter specifies what kind of comparison that is made for the evaluation of the mode condition.<br><br>For BSWM_EQUALS and BSWM_EQUALS_NOT, the BswMModeRequestPort port referenced by BswMConditionMode is compared with the value configured in BswMConditionValue for equality or not-equality.<br><br>For BSWM_EVENT_IS_SET and BSWM_EVENT_IS_CLEARED, the BswMEventRequestPort port referenced by BswMConditionMode is checked for being set or cleared (not-set). | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_EQUALS | | |
| | BSWM_EQUALS_NOT | | |
| | BSWM_EVENT_IS_CLEARED | | |
| | BSWM_EVENT_IS_SET | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMConditionMode [ECUC_BswM_00821] | | |
| **Parent Container** | BswMModeCondition | | |
| **Description** | This parameter references either a mode request port or an event request port. | | |
| **Multiplicity** | 1 | | |
| **Type** | Choice reference to [BswMEventRequestPort, BswMModeRequestPort] | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMConditionValue | 0..1 | This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to. |

**[SWS_BswM_00256] Configuration constraint for mode request comparisons and event request checking** ⌈The BswM shall reject configurations where `BswMConditionType` BSWM_EQUALS or BSWM_EQUALS_NOT are configured in conjunction with a `BswMEventRequestPort` port referenced by the `BswMConditionMode`. The BswM shall reject configurations where `BswMConditionType` BSWM_EVENT_IS_SET or BSWM_EVENT_IS_CLEARED are configured in conjunction with a `BswMModeRequestPort` port referenced by the `BswMConditionMode`.⌋ *(SRS_BSW_00167, SRS_ModeMgm_09177)*

### 10.2.6 BswMConditionValue

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00816] |
| **Container Name** | BswMConditionValue |
| **Parent Container** | BswMModeCondition |
| **Description** | This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to. |
| **Configuration Parameters** | |

| **Container Choices** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMBswMode | 0..1 | This container defines the value of a mode in the BSW. |
| BswMCompuScaleModeValue | 0..1 | This container contains parameters used to define a mode value. |
| BswMModeDeclaration | 0..1 | When the mode corresponds to a mode request or mode indication interface the mode is defined by a mode declaration. The mode declarations are defined in the SW-C Template and hence a foreign reference to the corresponding Mode Declaration is used. |

### 10.2.7 BswMBswMode

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00869] |
| **Container Name** | BswMBswMode |
| **Parent Container** | BswMConditionValue |
| **Description** | This container defines the value of a mode in the BSW. |
| **Configuration Parameters** | |

| | |
|---|---|
| **Name** | BswMBswRequestedMode [ECUC_BswM_00866] |
| **Parent Container** | BswMBswMode |
| **Description** | This parameter contains the symbolic name (as a string) of a certain mode/state that can be requested/indicated by the BSW modules. |
| **Multiplicity** | 1 |
| **Type** | EcucStringParamDef |
| **Default Value** | |
| **Regular Expression** | |

| | | | |
|---|---|---|---|
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.8 BswMModeDeclaration

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00868] |
| **Container Name** | BswMModeDeclaration |
| **Parent Container** | BswMConditionValue |
| **Description** | When the mode corresponds to a mode request or mode indication interface the mode is defined by a mode declaration. The mode declarations are defined in the SW-C Template and hence a foreign reference to the corresponding Mode Declaration is used. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMModeValueRef [ECUC_BswM_00864] | | |
| **Parent Container** | BswMModeDeclaration | | |
| **Description** | This is a foreign reference to the Mode Declaration used for the mode requests corresponding to this condition. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to MODE-DECLARATION | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.9 BswMEventRequestPort

| SWS Item | [ECUC_BswM_01052] | | |
|---|---|---|---|
| Container Name | BswMEventRequestPort | | |
| Parent Container | BswMArbitration | | |
| Description | Each instance of this container defines an event which can be sent to the BswM. Basic Software Modules may send these events to the BswM by calling the corresponding BswM C-API (for example: BswM_ComM_InitiateReset()). | | |
| Post-Build Variant Multiplicity | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Link time | – | |
| | Post-build time | – | |
| Configuration Parameters | | | |

| Name | BswMEventRequestProcessing [ECUC_BswM_01056] | | |
|---|---|---|---|
| Parent Container | BswMEventRequestPort | | |
| Description | This parameter defines if the processing of the mode arbitration shall be done immediately when an event request is received or if it shall be deferred to the processing of the main function of BswM. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | BSWM_DEFERRED | | |
| | BSWM_IMMEDIATE | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|---|---|---|
| Container Name | Multiplicity | Scope / Dependency |
| BswMEventRequest Source | 1 | This choice container specifies the source of the event request. The sender of the event can be another BSW Module, such as ComM. |

### 10.2.10 BswMModeRequestPort

| SWS Item | [ECUC_BswM_00805] |
|---|---|
| Container Name | BswMModeRequestPort |
| Parent Container | BswMArbitration |

| | | | |
|---|---|---|---|
| **Description** | Each instance of this container defines a mode request interface that is used to requests or indicate modes from/to the BswM. These interfaces are implemented as ports or as ordinary C-functions based upon if the request is made by an SW-C or a BSW module. There are different types of mode requests: 1. Mode requests from the SW-C:s 2. Mode Requests from other BSW modules such as the DCM. 3. State/mode indications from the RTE or other BSW modules such as the bus specific State Managers.<br><br>Note that the BswM treats all request and indications in the exact same way. | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Configuration Parameters** | | | |

| | | | |
|---|---|---|---|
| **Name** | BswMRequestProcessing [ECUC_BswM_00822] | | |
| **Parent Container** | BswMModeRequestPort | | |
| **Description** | This parameter defines if the processing of the mode arbitration shall be done immediately when a mode request is received or if it shall be deferred to the processing of the main function of BswM. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_DEFERRED | | |
| | BSWM_IMMEDIATE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMModeInitValue | 0..1 | This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. The initial mode value is defined by configuring either BswMBswModeInitValue or BswMCompuScaleModeValue. This container is optional. |
| BswMModeRequest Source | 1 | This choice container specifies the source of the mode request or state/mode indication. The requester of a mode can be both SW-C:s and other BSW Modules, such as the bus specific State Managers. |

### 10.2.11 BswMModeInitValue

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00928] |
| **Container Name** | BswMModeInitValue |
| **Parent Container** | BswMModeRequestPort |
| **Description** | This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. The initial mode value is defined by configuring either BswMBswModeInitValue or BswMCompuScaleModeValue. This container is optional. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMBswModeInitValue [ECUC_BswM_00932] | | |
| **Parent Container** | BswMModeInitValue | | |
| **Description** | This parameter defines the initial mode value that is used by BswM for the corresponding mode request after initialization. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucStringParamDef | | |
| **Default Value** | | | |
| **Regular Expression** | | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMCompuScaleModeValue | 0..1 | This container contains parameters used to define a mode value. |

### 10.2.12 BswMCompuScaleModeValue

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01039] |
| **Container Name** | BswMCompuScaleModeValue |
| **Parent Container** | BswMConditionValue, BswMModeInitValue |
| **Description** | This container contains parameters used to define a mode value. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMCompuConstText [ECUC_BswM_01041] | | |
| **Parent Container** | BswMCompuScaleModeValue | | |
| **Description** | The value of this parameter shall match the VT member of a CompuConst defined within the referenced CompuMethod (BswMCompuMethodRef). The interval value of the corresponding CompuScale shall be used as the mode request value. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucStringParamDef | | |
| **Default Value** | | | |
| **Regular Expression** | | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMCompuMethodRef [ECUC_BswM_01040] | | |
| **Parent Container** | BswMCompuScaleModeValue | | |
| **Description** | This is a foreign reference to the CompuMethod used for mode requests. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to COMPU-METHOD | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

**[SWS_BswM_CONSTR_00002]** ⌈The value of CompuMethod.category referenced by the foreign reference of `BswMCompuMethodRef` shall be TEXTTABLE.⌋ *(SRS_BSW_00167)*

### 10.2.13 BswMEventRequestSource

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01053] |
| **Container Name** | BswMEventRequestSource |
| **Parent Container** | BswMEventRequestPort |
| **Description** | This choice container specifies the source of the event request. The sender of the event can be another BSW Module, such as ComM. |
| **Configuration Parameters** | |

| Container Choices | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMComMInitiateReset | 0..1 | This is an indication from the ComM to signal a shutdown. |
| BswMDcmApplication UpdatedIndication | 0..1 | This is a request to update application data from the DCM. This container does not contain any parameters since there are no further configuration needed for this type of request. |
| BswMModeSwitchError Event | 0..1 | This is a notification that an error occurred because the partition containing mode users of the referenced PPort was restarted by the RTE. Because the Mode Machine Instance holding the current mode can reside on that terminated partition, the Mode Manager has to be informed about the loss of this partition. |
| BswMNmCarWakeUp Indication | 0..1 | This is an indication of a CarWakeup from the Nm. |
| BswMPartitionRestarted | 0..1 | This is a notification that an error occured because the partition containing the BswM was restarted by the RTE. The Mode Users may lie in another (still running) partition. So the BswM has to be informed that the start of its partition is no normal startup but a restart of a single partition. This information can be used inside the Rules. This notification has to be used by the Restart Task of the particular partition. |
| BswMSwitchAck Notification | 0..1 | This is a notification that a mode transition has been completed. |
| BswMWdgMRequest PartitionReset | 0..1 | This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.<br><br>**Tags:**<br>atp.Status=obsolete |

### 10.2.14 BswMModeRequestSource

| **SWS Item** | [ECUC_BswM_00856] |
|---|---|
| **Container Name** | BswMModeRequestSource |
| **Parent Container** | BswMModeRequestPort |
| **Description** | This choice container specifies the source of the mode request or state/mode indication. The requester of a mode can be both SW-C:s and other BSW Modules, such as the bus specific State Managers. |
| **Configuration Parameters** | |

| Container Choices | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMBswMode Notification | 0..1 | This is a mode request source emanating from another BSW Module. |
| BswMCanSMIndication | 0..1 | This is an indication of the current state of the CAN State Manager. |
| BswMComMIndication | 0..1 | This is an indication of the current communication mode of a channel in the Communication Manager. |

| | | |
|---|---|---|
| BswMComMPncRequest | 0..1 | This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager. |
| BswMDcmComMode Request | 0..1 | The source of the mode request is the Diagnostic Communication Manager. |
| BswMEcuMIndication | 0..1 | This is a notification of the current operation mode of the ECU State Manager. This container does not contain any parameters since there are no further configuration needed for this type of request. |
| BswMEcuMRUNRequest Indication | 0..1 | This is an indication of the current State of the RUN Request Protocol. |
| BswMEcuMWakeup Source | 0..1 | This is a notification of the current state of an ECU State Manager wakeup source. |
| BswMEthIfPortGroupLink StateChg | 0..1 | This is an indication from the EthIf if the link state of a Ethernet interface switch port group has changed. |
| BswMEthSMIndication | 0..1 | This is an indication of the current state of the Ethernet State Manager. |
| BswMFrSMIndication | 0..1 | This is an indication of the current state of the FlexRay State Manager. |
| BswMGenericRequest | 0..1 | This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers). |
| BswMJ1939Dcm BroadcastStatus | 0..1 | This is a notification of the desired broadcast status per network, triggered via DM13. |
| BswMJ1939NmIndication | 0..1 | This is an indication of the current state of the J1939 network management module. |
| BswMLinSMIndication | 0..1 | This is an indication of the current state of the LIN State Manager. |
| BswMLinSchedule Indication | 0..1 | This is an indication of the currently active LIN Schedule Table for a specific LIN Interface. |
| BswMLinTpMode Request | 0..1 | This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API. |
| BswMNmStateChange Notification | 0..1 | This is a notification from the Nm module that its state has changed. |
| BswMNvMJobMode Indication | 0..1 | Indicates the current status of the multiblock job. The job is identified via BswMNvmService. Possible values for this indication are the possible values of NvM_RequestResultType. |
| BswMNvMRequest | 0..1 | Via this Mode Request Source the NvM indicates the current status of the specified block. Possible Values are: NvM_RequestResultType NVM_REQ_OK NVM_REQ_NOT_OK NVM_REQ_PENDING NVM_REQ_INTEGRITY_FAILED NVM_REQ_BLOCK_SKIPPED NVM_REQ_NV_INVALIDATED NVM_REQ_CANCELED NVM_REQ_REDUNDANCY_FAILED NVM_REQ_RESTORED_FROM_ROM |
| BswMSdClientService CurrentState | 0..1 | Used by Service Discovery module to indicate current state of the Client Service (available/down). |

| | | |
|---|---|---|
| BswMSdConsumedEvent GroupCurrentState | 0..1 | Used by Service Discovery to indicate current status of the EventHandler (requested/released). |
| BswMSdEventHandler CurrentState | 0..1 | Used by Service Discovery to indicate current status of the EventHandler (requested/released). |
| BswMSoAdSoConMode Chg | 0..1 | This is an indication of the current state of the SoAd. |
| BswMSwcMode Notification | 0..1 | This is a mode switch notification associated with a RTE switch interface. |
| BswMSwcModeRequest | 0..1 | The source of the mode request is a SW Component. |
| BswMTimer | 0..1 | This is a timer which can be used for time dependent rules. This mode request port can be in one of three modes (depending on the state of the timer):<br>• BSWM_TIMER_STOPPED (intial) (The timer has been stopped by an action)<br>• BSWM_TIMER_STARTED (The timer has been started by an action)<br>• BSWM_TIMER_EXPIRED (The timer has expired) |

### 10.2.15 BswMBswModeNotification

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00926] |
| **Container Name** | BswMBswModeNotification |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is a mode request source emanating from another BSW Module. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMBswModeDeclarationGroupPrototypeRef [ECUC_BswM_00927] | | |
| **Parent Container** | BswMBswModeNotification | | |
| **Description** | This is a foreign reference to the Mode Declaration Group Prototype. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to MODE-DECLARATION-GROUP-PROTOTYPE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.16 BswMCanSMIndication

| SWS Item | [ECUC_BswM_00857] | | |
|---|---|---|---|
| Container Name | BswMCanSMIndication | | |
| Parent Container | BswMModeRequestSource | | |
| Description | This is an indication of the current state of the CAN State Manager. | | |
| Configuration Parameters | | | |

| Name | BswMCanSMChannelRef [ECUC_BswM_00870] | | |
|---|---|---|---|
| Parent Container | BswMCanSMIndication | | |
| Description | This is a reference to the CAN channel handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to ComMChannel | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.17 BswMComMIndication

| SWS Item | [ECUC_BswM_00880] | | |
|---|---|---|---|
| Container Name | BswMComMIndication | | |
| Parent Container | BswMModeRequestSource | | |
| Description | This is an indication of the current communication mode of a channel in the Communication Manager. | | |
| Configuration Parameters | | | |

| Name | BswMComMChannelRef [ECUC_BswM_00883] | | |
|---|---|---|---|
| Parent Container | BswMComMIndication | | |
| Description | This is a reference to the Communication Manager channel handle that the indication corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to ComMChannel | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |

| | |
|---|---|
| **Scope / Dependency** | scope: local |

**No Included Containers**

### 10.2.18 BswMComMInitiateReset

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01014] |
| **Container Name** | BswMComMInitiateReset |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is an indication from the ComM to signal a shutdown. |
| **Configuration Parameters** | |

**No Included Containers**

### 10.2.19 BswMComMPncRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00922] |
| **Container Name** | BswMComMPncRequest |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMPncRef [ECUC_BswM_00924] | | |
| **Parent Container** | BswMComMPncRequest | | |
| **Description** | This is a reference to the Communication Manager PNC handle of the Partial Network Cluster that the request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMPnc | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.20 BswMDcmApplicationUpdatedIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00925] |

| | |
|---|---|
| **Container Name** | BswMDcmApplicationUpdatedIndication |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is a request to update application data from the DCM. This container does not contain any parameters since there are no further configuration needed for this type of request. |
| **Configuration Parameters** | |

**No Included Containers**

### 10.2.21 BswMDcmComModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00863] |
| **Container Name** | BswMDcmComModeRequest |
| **Parent Container** | BswMModeRequestSource |
| **Description** | The source of the mode request is the Diagnostic Communication Manager. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMDcmComMChannelRef [ECUC_BswM_00969] | | |
| **Parent Container** | BswMDcmComModeRequest | | |
| **Description** | This is a reference from DcmModeRequest to the ComM channel that the indication corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.22 BswMEcuMIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01085] |
| **Container Name** | BswMEcuMIndication |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is a notification of the current operation mode of the ECU State Manager. This container does not contain any parameters since there are no further configuration needed for this type of request. |
| **Configuration Parameters** | |

| No Included Containers |
|---|

### 10.2.23 BswMEcuMRUNRequestIndication

| SWS Item | [ECUC_BswM_01086] |
|---|---|
| Container Name | BswMEcuMRUNRequestIndication |
| Parent Container | BswMModeRequestSource |
| Description | This is an indication of the current State of the RUN Request Protocol. |
| Configuration Parameters | |

| Name | BswMEcuMRUNRequestProtocolPort [ECUC_BswM_01087] | | |
|---|---|---|---|
| Parent Container | BswMEcuMRUNRequestIndication | | |
| Description | Identifies the EcuM State which is related to the mode request. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | BSWM_ECUM_STATE_POST_RUN | Port for POST_RUN State of EcuM. | |
| | BSWM_ECUM_STATE_RUN | Port for RUN State of EcuM. | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.24 BswMEcuMWakeupSource

| SWS Item | [ECUC_BswM_00904] |
|---|---|
| Container Name | BswMEcuMWakeupSource |
| Parent Container | BswMModeRequestSource |
| Description | This is a notification of the current state of an ECU State Manager wakeup source. |
| Configuration Parameters | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMWakeupSrcRef [ECUC_BswM_00905] | | |
| **Parent Container** | BswMEcuMWakeupSource | | |
| **Description** | This is a reference to the ECU State Manager Wakeup Source that the indication corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to EcuMWakeupSource | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.25 BswMEthIfPortGroupLinkStateChg

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01066] |
| **Container Name** | BswMEthIfPortGroupLinkStateChg |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is an indication from the EthIf if the link state of a Ethernet interface switch port group has changed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMEthIfSwitchPortGroupRef [ECUC_BswM_01067] | | |
| **Parent Container** | BswMEthIfPortGroupLinkStateChg | | |
| **Description** | This is a reference to the Ethernet Interface Switch Port Group that the indication corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to EthIfSwitchPortGroup | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.26 BswMEthSMIndication

| SWS Item | [ECUC_BswM_00860] | | |
|---|---|---|---|
| Container Name | BswMEthSMIndication | | |
| Parent Container | BswMModeRequestSource | | |
| Description | This is an indication of the current state of the Ethernet State Manager. | | |
| Configuration Parameters | | | |

| Name | BswMEthSMChannelRef [ECUC_BswM_00873] | | |
|---|---|---|---|
| Parent Container | BswMEthSMIndication | | |
| Description | This is a reference to the Ethernet channel handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to ComMChannel | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.27 BswMFrSMIndication

| SWS Item | [ECUC_BswM_00858] | | |
|---|---|---|---|
| Container Name | BswMFrSMIndication | | |
| Parent Container | BswMModeRequestSource | | |
| Description | This is an indication of the current state of the FlexRay State Manager. | | |
| Configuration Parameters | | | |

| Name | BswMFrSMChannelRef [ECUC_BswM_00872] | | |
|---|---|---|---|
| Parent Container | BswMFrSMIndication | | |
| Description | This is a reference to the FlexRay Cluster handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to ComMChannel | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.28 BswMGenericRequest

| SWS Item | [ECUC_BswM_00861] | | |
|---|---|---|---|
| Container Name | BswMGenericRequest | | |
| Parent Container | BswMModeRequestSource | | |
| Description | This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers). | | |
| Configuration Parameters | | | |

| Name | BswMModeRequesterId [ECUC_BswM_00874] | | |
|---|---|---|---|
| Parent Container | BswMGenericRequest | | |
| Description | This parameters identifies the different users of the generic mode request interface. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| Range | 0 .. 65535 | | |
| Default Value | | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | – | |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.29 BswMJ1939DcmBroadcastStatus

| SWS Item | [ECUC_BswM_00985] |
|---|---|
| Container Name | BswMJ1939DcmBroadcastStatus |
| Parent Container | BswMModeRequestSource |
| Description | This is a notification of the desired broadcast status per network, triggered via DM13. |
| Configuration Parameters | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939DcmChannelRef [ECUC_BswM_00988] | | |
| **Parent Container** | BswMJ1939DcmBroadcastStatus | | |
| **Description** | Reference to the communication channel which is affected by this mode request. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

**[SWS_BswM_00250]** ⌈Since NetworkMask (in the `BswM_J1939DcmBroadcastStatus` API) is 16 bits wide, the BswM shall reject configurations where the `ComMChannel.ComMChannelId` referenced by a `BswMJ1939DcmBroadcastStatus.BswMJ1939DcmChannelRef` parameter is greater than 15.⌋*(SRS_ModeMgm_09228)*

### 10.2.30 BswMJ1939NmIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00966] |
| **Container Name** | BswMJ1939NmIndication |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is an indication of the current state of the J1939 network management module. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939NmChannelRef [ECUC_BswM_00967] | | |
| **Parent Container** | BswMJ1939NmIndication | | |
| **Description** | This is a reference to the J1939Nm channel handle that the mode request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939NmNodeRef [ECUC_BswM_00997] | | |
| **Parent Container** | BswMJ1939NmIndication | | |
| **Description** | This is a reference to the node that the mode request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to J1939NmNode | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.31 BswMLinSMIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00859] |
| **Container Name** | BswMLinSMIndication |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is an indication of the current state of the LIN State Manager. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMLinSMChannelRef [ECUC_BswM_00871] | | |
| **Parent Container** | BswMLinSMIndication | | |
| **Description** | This is a reference to the LIN channel handle that the mode request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.32 BswMLinScheduleIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00885] |

| Container Name | BswMLinScheduleIndication | | |
|---|---|---|---|
| Parent Container | BswMModeRequestSource | | |
| Description | This is an indication of the currently active LIN Schedule Table for a specific LIN Interface. | | |
| Configuration Parameters | | | |

| Name | BswMLinScheduleRef [ECUC_BswM_00886] | | |
|---|---|---|---|
| Parent Container | BswMLinScheduleIndication | | |
| Description | This is a reference to the LIN Schedule Table handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to LinSMSchedule | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| Name | BswMLinSMChannelRef [ECUC_BswM_01028] | | |
|---|---|---|---|
| Parent Container | BswMLinScheduleIndication | | |
| Description | This is a reference to the LIN channel handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to ComMChannel | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.33 BswMLinTpModeRequest

| SWS Item | [ECUC_BswM_00914] |
|---|---|
| Container Name | BswMLinTpModeRequest |
| Parent Container | BswMModeRequestSource |
| Description | This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API. |
| Configuration Parameters | |

| | | | |
|---|---|---|---|
| **Name** | BswMLinTpChannelRef [ECUC_BswM_00915] | | |
| **Parent Container** | BswMLinTpModeRequest | | |
| **Description** | This is a reference to the LIN Interface Channel that the mode request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.34 BswMModeSwitchErrorEvent

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00990] |
| **Container Name** | BswMModeSwitchErrorEvent |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is a notification that an error occurred because the partition containing mode users of the referenced PPort was restarted by the RTE. Because the Mode Machine Instance holding the current mode can reside on that terminated partition, the Mode Manager has to be informed about the loss of this partition. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMRteSwitchPortRef [ECUC_BswM_01030] | | |
| **Parent Container** | BswMModeSwitchErrorEvent | | |
| **Description** | This is a reference to the BswMSwitchPort. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMSwitchPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

**[SWS_BswM_00259]** ⌈`BswMModeSwitchErrorEvent` specifies a SwcModeManagerErrorEvent, which the BswM shall create in its SWCD. The ModeDeclarationGroupPrototype which is referenced by the SwcModeManagerErrorEvent.modeGroup shall correspond to the one used by the ModeSwitchInterface which is referenced by the `BswMSwitchPort` that is configured by the `BswMRteSwitchPortRef` in `BswMModeSwitchErrorEvent`. The BswM shall create an associated runnable which will arbitrate the SwcModeManagerErrorEvent.⌋ *(SRS_ModeMgm_09182)*

### 10.2.35 BswMNmCarWakeUpIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01075] |
| **Container Name** | BswMNmCarWakeUpIndication |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is an indication of a CarWakeup from the Nm. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMNmChannelRef [ECUC_BswM_01049] | | |
| **Parent Container** | BswMNmCarWakeUpIndication | | |
| **Description** | This is a reference to the channel handle that the indication corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| |
|---|
| **No Included Containers** |

### 10.2.36 BswMNmStateChangeNotification

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01081] |
| **Container Name** | BswMNmStateChangeNotification |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is a notification from the Nm module that its state has changed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMNmChannelRef [ECUC_BswM_01082] | | |
| **Parent Container** | BswMNmStateChangeNotification | | |
| **Description** | This is a reference to the channel handle that this notification corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.37 BswMNvMJobModeIndication

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00956] |
| **Container Name** | BswMNvMJobModeIndication |
| **Parent Container** | BswMModeRequestSource |
| **Description** | Indicates the current status of the multiblock job. The job is identified via BswMNvmService. Possible values for this indication are the possible values of NvM_RequestResultType. |
| **Configuration Parameters** | |

| | | |
|---|---|---|
| **Name** | BswMNvmService [ECUC_BswM_00957] | |
| **Parent Container** | BswMNvMJobModeIndication | |
| **Description** | Identifies the Nvm job which is related to the mode request. | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | NvmCancelWriteAll | corresponds to multi block service NvM_CancelWriteAll |
| | NvmFirstInitAll | corresponds to multi block service NvM_FirstInitAll |
| | NvmReadAll | corresponds to multi block service NvM_ReadAll |
| | NvmValidateAll | corresponds to multi block service NvM_ValidateAll |
| | NvmWriteAll | corresponds to multi block service NvM_WriteAll |
| **Post-Build Variant Value** | false | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.38 BswMNvMRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00890] |
| **Container Name** | BswMNvMRequest |
| **Parent Container** | BswMModeRequestSource |
| **Description** | Via this Mode Request Source the NvM indicates the current status of the specified block. Possible Values are: NvM_RequestResultType NVM_REQ_OK NVM_REQ_NOT_OK NVM_REQ_PENDING NVM_REQ_INTEGRITY_FAILED NVM_REQ_BLOCK_SKIPPED NVM_REQ_NV_INVALIDATED NVM_REQ_CANCELED NVM_REQ_REDUNDANCY_FAILED NVM_REQ_RESTORED_FROM_ROM |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMNvMBlockRef [ECUC_BswM_00891] | | |
| **Parent Container** | BswMNvMRequest | | |
| **Description** | This is a reference to the NvM Block Descriptor that the request corresponds to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to NvMBlockDescriptor | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.39 BswMPartitionRestarted

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00989] |
| **Container Name** | BswMPartitionRestarted |
| **Parent Container** | BswMEventRequestSource |

| | |
|---|---|
| **Description** | This is a notification that an error occured because the partition containing the BswM was restarted by the RTE. The Mode Users may lie in another (still running) partition. So the BswM has to be informed that the start of its partition is no normal startup but a restart of a single partition. This information can be used inside the Rules. This notification has to be used by the Restart Task of the particular partition. |
| **Configuration Parameters** | |

| |
|---|
| **No Included Containers** |

### 10.2.40 BswMSdClientServiceCurrentState

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01011] |
| **Container Name** | BswMSdClientServiceCurrentState |
| **Parent Container** | BswMModeRequestSource |
| **Description** | Used by Service Discovery module to indicate current state of the Client Service (available/down). |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdClientMethodsRef [ECUC_BswM_01009] | | |
| **Parent Container** | BswMSdClientServiceCurrentState | | |
| **Description** | This is a reference to a client service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdClientService | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| |
|---|
| **No Included Containers** |

### 10.2.41 BswMSdConsumedEventGroupCurrentState

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01012] |
| **Container Name** | BswMSdConsumedEventGroupCurrentState |
| **Parent Container** | BswMModeRequestSource |
| **Description** | Used by Service Discovery to indicate current status of the EventHandler (requested/released). |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdConsumedEventGroupRef [ECUC_BswM_01010] | | |
| **Parent Container** | BswMSdConsumedEventGroupCurrentState | | |
| **Description** | This is a reference to an eventGroup that is defined within a client service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdConsumedEventGroup | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.42 BswMSdEventHandlerCurrentState

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01013] |
| **Container Name** | BswMSdEventHandlerCurrentState |
| **Parent Container** | BswMModeRequestSource |
| **Description** | Used by Service Discovery to indicate current status of the EventHandler (requested/released). |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdEventHandlerRef [ECUC_BswM_01008] | | |
| **Parent Container** | BswMSdEventHandlerCurrentState | | |
| **Description** | This is a reference to an event handler that is defined within a server service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdEventHandler | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.43 BswMSoAdSoConModeChg

| SWS Item | [ECUC_BswM_01091] |
|---|---|
| Container Name | BswMSoAdSoConModeChg |
| Parent Container | BswMModeRequestSource |
| Description | This is an indication of the current state of the SoAd. |
| Post-Build Variant Multiplicity | false |
| Configuration Parameters | |

| Name | BswMSoAdSocketIdRef [ECUC_BswM_01092] | | |
|---|---|---|---|
| Parent Container | BswMSoAdSoConModeChg | | |
| Description | This is a reference to the SoAd socket ID handle that the mode request corresponds to. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to SoAdSocketConnection | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.44 BswMSwcModeNotification

| SWS Item | [ECUC_BswM_00892] |
|---|---|
| Container Name | BswMSwcModeNotification |
| Parent Container | BswMModeRequestSource |
| Description | This is a mode switch notification associated with a RTE switch interface. |
| Configuration Parameters | |

| Name | BswMSwcModeNotificationModeDeclarationGroupPrototypeRef [ECUC_BswM_00893] |
|---|---|
| Parent Container | BswMSwcModeNotification |
| Description | This is a foreign reference to the ModeDeclarationGroupPrototype. |
| Multiplicity | 1 |
| Type | Foreign reference to MODE-DECLARATION-GROUP-PROTOTYPE |
| Post-Build Variant Value | false |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.45 BswMSwcModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00862] |
| **Container Name** | BswMSwcModeRequest |
| **Parent Container** | BswMModeRequestSource |
| **Description** | The source of the mode request is a SW Component. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSwcModeRequestVariableDataPrototypeRef [ECUC_BswM_01046] | | |
| **Parent Container** | BswMSwcModeRequest | | |
| **Description** | This is a reference to the VariableDataPrototype. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to VARIABLE-DATA-PROTOTYPE | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

**[SWS_BswM_00236]** ⌈The BswM shall only accept configurations where the VariableDataPrototype, which is referenced by a `BswMSwcModeRequest`, belongs to a SenderReceiverInterface.⌋ *(SRS_ModeMgm_09179)*

### 10.2.46 BswMSwitchAckNotification

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01083] |
| **Container Name** | BswMSwitchAckNotification |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is a notification that a mode transition has been completed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSwitchPortRef [ECUC_BswM_01084] | | |
| **Parent Container** | BswMSwitchAckNotification | | |
| **Description** | References the switch port which will receive the notification. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMSwitchPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

**[SWS_BswM_00284]** ⌈For `BswMSwitchAckNotification`, if the referenced `BswMSwitchPort` is configured with `BswMModeSwitchInterfaceRef` without a `BswMSchMModeDeclarationGroupRef`: This specifies a ModeSwitchedAckEvent, which the BswM shall create in its SWCD. The BswM shall create an associated runnable; the C definition of this runnable/schedulable entity shall arbitrate the ModeSwitchedAckEvent.⌋*()*

**[SWS_BswM_00285]** ⌈For `BswMSwitchAckNotification`, if the referenced `BswMSwitchPort` is configured with `BswMSchMModeDeclarationGroupRef`: This specifies a BswModeSwitchedAckEvent, which the BswM shall create in its BSWMD. The BswM shall create an associated runnable; the C definition of this runnable/schedulable entity shall arbitrate the BswModeSwitchedAckEvent.⌋*()*

### 10.2.47 BswMTimer

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01058] |
| **Container Name** | BswMTimer |
| **Parent Container** | BswMModeRequestSource |
| **Description** | This is a timer which can be used for time dependent rules. This mode request port can be in one of three modes (depending on the state of the timer):<br>• BSWM_TIMER_STOPPED (intial) (The timer has been stopped by an action)<br>• BSWM_TIMER_STARTED (The timer has been started by an action)<br>• BSWM_TIMER_EXPIRED (The timer has expired) |

**Configuration Parameters**

**No Included Containers**

### 10.2.48 BswMWdgMRequestPartitionReset

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00916] (Obsolete) |
| **Container Name** | BswMWdgMRequestPartitionReset |
| **Parent Container** | BswMEventRequestSource |
| **Description** | This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.<br><br>**Tags:**<br>atp.Status=obsolete |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMWdgMRequestPartitionResetRef [ECUC_BswM_00917] (Obsolete) | | |
| **Parent Container** | BswMWdgMRequestPartitionReset | | |
| **Description** | This is a reference to the partition that shall be reset.<br><br>**Tags:**<br>atp.Status=obsolete | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to EcucPartition | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.49 BswMRule

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00806] | | |
| **Container Name** | BswMRule | | |
| **Parent Container** | BswMArbitration | | |
| **Description** | Each instance of this container describes a BswM arbitration rule. The rule either consists of a simple mode condition or a more complex logical expression. This container also references the action lists that shall be invoked when the rule is evaluated to True or False. | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Configuration Parameters** | | | |

| | | | |
|---|---|---|---|
| **Name** | BswMNestedExecutionOnly [ECUC_BswM_00935] | | |
| **Parent Container** | BswMRule | | |
| **Description** | This parameter defines for its related Rule if the Rule is an Independent rule or a Subordinate rule;<br><br>false: an Independent rule, i.e. to be evaluated each time applicable (both as standalone Rule driven by its own BswMModeRequestSource and when referenced by another Rule).<br><br>true: a Subordinated rule, to be evaluated ONLY as a result of being referenced in one or more Action Lists. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMRuleInitState [ECUC_BswM_00888] | | |
| **Parent Container** | BswMRule | | |
| **Description** | This parameter is a part of the reset/initialization behavior of BswM. Action lists are executed when the result of a rule evaluation have changed since the last evaluation. This parameter defines the "previous evaluation result" of a rule to be used after initialization of the BswM.<br><br>If this parameter is set to BSWM_UNDEFINED, the evaluation result is always treated as changed at the first evaluation of the rule after initialization.<br><br>If this parameter is set to BSWM_TRUE, the evaluation result is treated as changed if the rule is evaluated to false.<br><br>If this parameter is set to BSWM_FALSE, the evaluation result is treated as changed if the rule is evaluated to true. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_FALSE | | |
| | BSWM_TRUE | | |
| | BSWM_UNDEFINED | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |

| | |
|---|---|
| **Scope / Dependency** | scope: local |

| | | | |
|---|---|---|---|
| **Name** | BswMRuleExpressionRef [ECUC_BswM_00819] | | |
| **Parent Container** | BswMRule | | |
| **Description** | This is a reference to the logical expression that is evaluated for each rule. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMLogicalExpression | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMRuleFalseActionList [ECUC_BswM_00818] | | |
| **Parent Container** | BswMRule | | |
| **Description** | This is a reference to the action list that shall be executed when the rule is evaluated to False | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Reference to BswMActionList | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **Name** | BswMRuleTrueActionList [ECUC_BswM_00817] |
| **Parent Container** | BswMRule |
| **Description** | This is a reference to the action list that shall be executed when the rule is evaluated to True |
| **Multiplicity** | 0..1 |
| **Type** | Reference to BswMActionList |
| **Post-Build Variant Multiplicity** | true |
| **Post-Build Variant Value** | true |

| | | | |
|---|---|---|---|
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.50 BswMDataTypeMappingSets

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00936] |
| **Container Name** | BswMDataTypeMappingSets |
| **Parent Container** | BswMConfig |
| **Description** | Collection of references to DataTypeMappingSet. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMDataTypeMappingSetRef [ECUC_BswM_00937] | | |
| **Parent Container** | BswMDataTypeMappingSets | | |
| **Description** | Reference to DataTypeMappingSet. | | |
| **Multiplicity** | 1..* | | |
| **Type** | Foreign reference to DATA-TYPE-MAPPING-SET | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.51 BswMModeControl

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00802] |

| Container Name | BswMModeControl |
|---|---|
| Parent Container | BswMConfig |
| Description | This container includes all configuration sub-containers and parameters related to the mode control functionality of the BswM. |
| Configuration Parameters | |

| Included Containers | | |
|---|---|---|
| Container Name | Multiplicity | Scope / Dependency |
| BswMAction | 0..* | Each container of this type defines an action. These actions can be part of one or several action lists. |
| BswMActionList | 0..* | Each instance of this container defines an action list that is invoked based on the BswM Rules. An action list contains a list of numbered action items to be processed. An action list can also include other action lists. |
| BswMRteModeRequestPort | 0..* | This container defines a mode request port which the BswM may utilize to send a mode request to a SW-C which is acting as a mode-manager. If this container is referenced by a BswMRteModeRequest, the BswM shall create a corresponding PPort in its service description. |
| BswMSwitchPort | 0..* | Represents an output mode-switch port to be generated by the BswM. If BswMModeSwitchInterfaceRef is configured then a PPortPrototype is generated in the SWCD. If BswMSchMModeDeclarationGroupRef is configured then a ModeDeclarationGroupPrototype is generated in the ProvidedModeGroups of the BSWMD. If both BswMModeSwitchInterfaceRef and BswMSchMModeDeclarationGroupRef are configured then an SwcBswSynchronizedModeGroupPrototype is also generated in the BSWMD (see Chapter 6.11 of the BSW Module Description Template SWS and EXP ModemanagementGuide).. |

### 10.2.52 BswMAction

| SWS Item | [ECUC_BswM_00810] | | |
|---|---|---|---|
| Container Name | BswMAction | | |
| Parent Container | BswMModeControl | | |
| Description | Each container of this type defines an action. These actions can be part of one or several action lists. | | |
| Post-Build Variant Multiplicity | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Link time | – | |
| | Post-build time | – | |
| Configuration Parameters | | | |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMAvailableActions | 1 | Choice container including the available actions to be used in the action lists. |

### 10.2.53 BswMAvailableActions

| SWS Item | [ECUC_BswM_00826] |
|---|---|
| Container Name | BswMAvailableActions |
| Parent Container | BswMAction |
| Description | Choice container including the available actions to be used in the action lists. |
| Configuration Parameters | |

| Container Choices | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMClearEventRequest | 0..1 | This container contains a reference to a BswMEventRequestPort which will be cleared (i.e. set to CLEAR state) when this action is executed. |
| BswMComMAllowCom | 0..1 | This container includes all parameters for the action to allow or to block communication for a ComM Channel. ComM_CommunicationAllowed is called when this action is configured. |
| BswMComMMode Limitation | 0..1 | This container includes all parameters related to a limitation of communication mode for a ComM Channel. ComM_LimitChannelToNoComMode is called when this action is configured. |
| BswMComMModeSwitch | 0..1 | This container includes all parameters related to a switch of communication mode for a ComM User. ComM_RequestComMode is called when this action is configured. |
| BswMCoreHaltMode | 0..1 | This container includes all parameters related to a switch of the activation state of core Halt. |
| BswMDeadlineMonitoring Control | 0..1 | This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM. |
| BswMEcuMDriverInitList BswM | 0..1 | This container defines the action to trigger an EcuM driver initialization list. |
| BswMEcuMGoDownHalt Poll | 0..1 | This container defines the action to trigger the EcuM_GoDownHaltPoll from BswM. |
| BswMEcuMSelect ShutdownTarget | 0..1 | This container defines the shutdown target. |
| BswMEcuMStateSwitch | 0..1 | This container defines the action to switch a State of the ECU State Manager. |
| BswMEthIfStartAllPorts | 0..1 | This container defines the action to call EthIf_StartAllPorts from BswM and thus to start all affected EthSwtPorts. |

| | | |
|---|---|---|
| BswMEthIfSwitchPort GroupRequestMode | 0..1 | This container includes all parameters related to requesting a mode for the EthIfSwtPortGroup.<br><br>The EthIf_SwitchPortGroupRequestMode API is called when this action is executed. |
| BswMFrSMAllSlots | 0..1 | This container includes all parameter(s) for the action to request an exit from Flexray KeySlotOnlyMode. FrSM_AllSlots is called when this action is executed. |
| BswMIdsMBlockState ChangeRequest | 0..1 | This container defines the action to switch a BlockState of the IdsM.<br><br>**Tags:**<br>atp.Status=draft |
| BswMJ1939DcmState Switch | 0..1 | This container includes all parameters related to a switch of the J1939 Diagnostic Communication Managers network state for a J1939 node. J1939Dcm_SetState is called when this action is configured. |
| BswMJ1939RmState Switch | 0..1 | This container includes all parameters related to a switch of the J1939 Request Managers network state for a J1939 node. J1939Rm_SetState is called when this action is configured. |
| BswMLinScheduleSwitch | 0..1 | This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured.<br><br>The configuration for the "network" parameter can be accessed via the reference LinSMComMNetworkHandleRef contained in the parent container LinSMChannel of the container referenced by BswMLinScheduleRef. |
| BswMNMControl | 0..1 | This container includes all parameters related to enabling and disabling of Network Management communication. Disabling of NM communication can be requested by DCM. Nm_EnableCommunication or Nm_DisableCommunication is called when this action is configured. |
| BswMPduGroupSwitch | 0..1 | This container includes references to the PDU groups that shall be enabled and disabled. |
| BswMPduRouterControl | 0..1 | This container includes all parameters related to enabling and disabling of routing of Routing Path Groups in the PDU Router. PduR_EnableRouting or PduR_DisableRouting is called when this action is configured. |
| BswMRteModeRequest | 0..1 | This container defines a mode request that the BswM may send to a SW-C which is acting as a mode-manager. RTE_Write is called when this action is configured. |
| BswMRteStart | 0..1 | This container defines the action to start the Rte from BswM. |
| BswMRteStop | 0..1 | This container defines the action to stop the Rte from BswM |

| | | |
|---|---|---|
| BswMRteSwitch | 0..1 | This container defines a mode switch indication that the BswM provides to the SW-C that need to be notified about the mode switch. RTE_Switch is called when this action is configured. |
| BswMSchMSwitch | 0..1 | This container defines a mode switch indication that the BswM provides to the SW-C that need to be notified about the mode switch. SchM_Switch is called when this action is configured. |
| BswMSdClientService ModeRequest | 0..1 | This container includes all parameters related to the selection of an client service of Sd. Sd_ClientServiceSetState is called when this action is configured. |
| BswMSdConsumedEvent GroupModeRequest | 0..1 | This container includes all parameters related to the selection of a consumed EventGroup of Sd. Sd_ConsumedEventGroupSetState is called when this action is configured. |
| BswMSdServerService ModeRequest | 0..1 | This container includes all parameters related to the selection of a server service of Sd. Sd_ServerServiceSetState is called when this action is configured. |
| BswMSdServiceGroup Switch | 0..1 | This container includes references to the SdServiceGroups that shall be enabled and disabled. |
| BswMSwitchIPduMode | 0..1 | This container includes all parameters related to the selection of the transmission mode an I-PDU to be sent by COM. Com_SwitchIpduTxMode is called when this action is configured. |
| BswMTimerControl | 0..1 | This container includes all parameters for the action to start or to stop a timer. |
| BswMTriggerIPduSend | 0..1 | This container includes all parameters related to the triggering of an I-PDU to be sent by COM. Com_TriggerIPDUSend is called when this action is configured. |
| BswMUserCallout | 0..1 | This container includes all details needed for a user defined function call. |

### 10.2.54 BswMClearEventRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01054] |
| **Container Name** | BswMClearEventRequest |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container contains a reference to a BswMEventRequestPort which will be cleared (i.e. set to CLEAR state) when this action is executed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMClearEventRequestPortRef [ECUC_BswM_01055] | | |
| **Parent Container** | BswMClearEventRequest | | |
| **Description** | This parameter references the BswMEventRequestPort which will be cleared. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMEventRequestPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.55 BswMComMAllowCom

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00909] |
| **Container Name** | BswMComMAllowCom |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters for the action to allow or to block communication for a ComM Channel. ComM_CommunicationAllowed is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMComAllowed [ECUC_BswM_00918] | | |
| **Parent Container** | BswMComMAllowCom | | |
| **Description** | The parameter BswMComMAllowChannelRef refers to a channel which will allow or block communication using the function ComM_CommunicationAllowed().<br><br>This parameter corresponds to the parameter "Allowed" of the function ComM_CommunicationAllowed(). | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMAllowChannelRef [ECUC_BswM_00912] | | |
| **Parent Container** | BswMComMAllowCom | | |
| **Description** | This is a reference to the ComM Channel for which communication shall be allowed or blocked.<br><br>This reference corresponds to the parameter "Channel" of the function ComM_CommunicationAllowed(). | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.56 BswMComMModeLimitation

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00908] |
| **Container Name** | BswMComMModeLimitation |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to a limitation of communication mode for a ComM Channel. ComM_LimitChannelToNoComMode is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMLimitMode [ECUC_BswM_00910] | | |
| **Parent Container** | BswMComMModeLimitation | | |
| **Description** | The function ComM_LimitChannelToNoComMode() takes in this boolean parameter to limit the channel's com mode to no-com mode.<br><br>This parameter corresponds to the parameter "Status" of the function ComM_LimitChannelToNoComMode. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |

| | |
|---|---|
| **Scope / Dependency** | scope: local |

| | | | |
|---|---|---|---|
| **Name** | BswMComMLimitChannelRef [ECUC_BswM_00911] | | |
| **Parent Container** | BswMComMModeLimitation | | |
| **Description** | This is a reference to the ComM channel for which the communication mode should be limited.<br><br>This reference corresponds to the parameter "Channel" of the function ComM_LimitChannelToNoComMode. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| |
|---|
| **No Included Containers** |

### 10.2.57 BswMComMModeSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00831] |
| **Container Name** | BswMComMModeSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to a switch of communication mode for a ComM User. ComM_RequestComMode is called when this action is configured. |
| **Configuration Parameters** | |

| | | |
|---|---|---|
| **Name** | BswMComMRequestedMode [ECUC_BswM_00840] | |
| **Parent Container** | BswMComMModeSwitch | |
| **Description** | This parameter specifies if the requested communication mode.<br><br>This parameter corresponds to the parameter "ComMode" of the function ComM_RequestComMode. | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BSWM_COMM_FULL_COMMUNICATION | |
| | BSWM_COMM_NO_COMMUNICATION | |
| **Post-Build Variant Value** | false | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMUserRef [ECUC_BswM_00841] | | |
| **Parent Container** | BswMComMModeSwitch | | |
| **Description** | This is a reference to the ComM User that is associated to the Communication channel for which the communication mode should be requested.<br><br>This reference corresponds to the parameter "User" of the function ComM_RequestComMode. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMUser | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.58 BswMCoreHaltMode

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00970] |
| **Container Name** | BswMCoreHaltMode |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to a switch of the activation state of core Halt. |
| **Configuration Parameters** | |

| | |
|---|---|
| **Name** | BswMCoreHaltActivationState [ECUC_BswM_00972] |
| **Parent Container** | BswMCoreHaltMode |
| **Description** | Different possibilities are offered depending on the OS implementation and the CPU HW. The HALT modes addressed by this parameter are defined as names (strings) in the OS implementation. Different implementation may implement different HALT modes and subsequently different names. |
| **Multiplicity** | 1 |
| **Type** | EcucStringParamDef |
| **Default Value** | |

| | | | |
|---|---|---|---|
| **Regular Expression** | | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMTargetCoreRef [ECUC_BswM_00971] | | |
| **Parent Container** | BswMCoreHaltMode | | |
| **Description** | This is a reference to the core on which the Core Halt process must be influenced. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to EcucCoreDefinition | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.59 BswMDeadlineMonitoringControl

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00830] |
| **Container Name** | BswMDeadlineMonitoringControl |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM. |
| **Configuration Parameters** | |

| | |
|---|---|
| **Name** | BswMDisabledDMPduGroupRef [ECUC_BswM_00852] |
| **Parent Container** | BswMDeadlineMonitoringControl |
| **Description** | This is a reference to a PDU Group for which the Deadline Monitoring should be disabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_DisableReceptionDM. |
| **Multiplicity** | 0..* |
| **Type** | Symbolic name reference to ComIPduGroup |
| **Post-Build Variant Multiplicity** | true |
| **Post-Build Variant Value** | true |

| | | | |
|---|---|---|---|
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEnabledDMPduGroupRef [ECUC_BswM_00851] | | |
| **Parent Container** | BswMDeadlineMonitoringControl | | |
| **Description** | This is a reference to a PDU Group for which the Deadline Monitoring should be enabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_EnableReceptionDM. | | |
| **Multiplicity** | 0..* | | |
| **Type** | Symbolic name reference to ComIPduGroup | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

**[SWS_BswM_CONSTR_00003]** ⌈The BswM shall reject configurations where a `BswMDeadlineMonitoringControl` container has a `BswMDisabledDMPduGroupRef` and a `BswMEnabledDMPduGroupRef` which reference the same PDU Group.⌋*()*

**[SWS_BswM_00274]** ⌈When a `BswMDeadlineMonitoringControl` action is executed, the BswM shall call `Com_EnableReceptionDM` for each `BswMEnabledDMPduGroupRef`, and call `Com_DisableReceptionDM` for each `BswMDisabledDMPduGroupRef`. The ordering of these calls to Com is undefined.⌋*()*

Note: If a strict ordering of the calls to `Com_EnableReceptionDM`, or `Com_DisableReceptionDM` is required, then this can be achieved by configuring individual actions (`BswMDeadlineMonitoringControl`, each with just a single BswM*PduGroupRef) within an ordered action list.

### 10.2.60 BswMEcuMDriverInitListBswM

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01064] |
| **Container Name** | BswMEcuMDriverInitListBswM |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines the action to trigger an EcuM driver initialization list. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMDriverInitListBswMRef [ECUC_BswM_01065] | | |
| **Parent Container** | BswMEcuMDriverInitListBswM | | |
| **Description** | This is a reference to the EcuM EcuMDriverInitListBswM container which represents the driver init list to be triggered. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to EcuMDriverInitListBswM | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

The EcuM (flex) can be configured with driver initialization lists (EcuMDriverInitListBswM) which may then be called by the BswM.

**[SWS_BswM_00269]** ⌈When a `BswMEcuMDriverInitListBswM` action is executed, the BswM shall call the `EcuM_AL_DriverInitBswM_<EcuMDriverInitListBswM.shortName>(void)` function which is provided by the EcuM.⌋*(SRS_ModeMgm_09180)*

### 10.2.61 BswMEcuMGoDownHaltPoll

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00963] |
| **Container Name** | BswMEcuMGoDownHaltPoll |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines the action to trigger the EcuM_GoDownHaltPoll from BswM. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMUserIdRef [ECUC_BswM_00964] | | |
| **Parent Container** | BswMEcuMGoDownHaltPoll | | |
| **Description** | This is a reference to a EcuM UserId. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to EcuMFlexUserConfig | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.62 BswMEcuMSelectShutdownTarget

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00961] |
| **Container Name** | BswMEcuMSelectShutdownTarget |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines the shutdown target. |
| **Configuration Parameters** | |

| | | |
|---|---|---|
| **Name** | BswMEcuMShutdownTarget [ECUC_BswM_00993] | |
| **Parent Container** | BswMEcuMSelectShutdownTarget | |
| **Description** | This parameter contains the shutdown target that the BswM selects at the EcuM. | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BSWM_ECUM_SHUTDOWN_TARGET_OFF | |
| | BSWM_ECUM_SHUTDOWN_TARGET_RESET | In case the configuration parameter BswMEcuMShutdownTarget is set to BSWM_ECUM_SHUTDOWN_TARGET_RESET the configuration parameter BswMEcuMResetModeRef shall exist and contain a valid reference to a EcuM reset mode. |
| | BSWM_ECUM_SHUTDOWN_TARGET_SLEEP | In case the configuration parameter BswMEcuMShutdownTarget is set to BSWM_ECUM_SHUTDOWN_TARGET_SLEEP the configuration parameter BswMEcuMSleepModeRef shall exist and contain a valid reference to a EcuM sleep mode. |
| **Post-Build Variant Value** | false | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMResetModeRef [ECUC_BswM_00994] | | |
| **Parent Container** | BswMEcuMSelectShutdownTarget | | |
| **Description** | This is a reference to a reset mode. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Symbolic name reference to EcuMResetMode | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMSleepModeRef [ECUC_BswM_00962] | | |
| **Parent Container** | BswMEcuMSelectShutdownTarget | | |
| **Description** | This is a reference to a sleep mode. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Symbolic name reference to EcuMSleepMode | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| No Included Containers |
|---|

### 10.2.63 BswMEcuMStateSwitch

| SWS Item | [ECUC_BswM_01045] |
|---|---|
| Container Name | BswMEcuMStateSwitch |
| Parent Container | BswMAvailableActions |
| Description | This container defines the action to switch a State of the ECU State Manager. |
| Configuration Parameters | |

| Name | BswMEcuMState [ECUC_BswM_01044] | | |
|---|---|---|---|
| Parent Container | BswMEcuMStateSwitch | | |
| Description | This parameter corresponds to the parameter "State" of the function EcuM_ SetState (). | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | BSWM_ECUM_STATE_POST_RUN | | |
| | BSWM_ECUM_STATE_RUN | | |
| | BSWM_ECUM_STATE_SHUTDOWN | | |
| | BSWM_ECUM_STATE_SLEEP | | |
| | BSWM_ECUM_STATE_STARTUP | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| Scope / Dependency | scope: local | | |

| No Included Containers |
|---|

### 10.2.64 BswMEthIfStartAllPorts

| SWS Item | [ECUC_BswM_01088] |
|---|---|
| Container Name | BswMEthIfStartAllPorts |
| Parent Container | BswMAvailableActions |
| Description | This container defines the action to call EthIf_StartAllPorts from BswM and thus to start all affected EthSwtPorts. |
| Configuration Parameters | |

| No Included Containers |
|---|

**[SWS_BswM_00286]** ⌈If the action `BswMEthIfStartAllPorts` is configured, the function `EthIf_StartAllPorts(void)` shall be called by the BswM when the action is executed.⌋*()*

### 10.2.65 BswMEthIfSwitchPortGroupRequestMode

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01068] |
| **Container Name** | BswMEthIfSwitchPortGroupRequestMode |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to requesting a mode for the EthIfSwtPortGroup.<br><br>The EthIf_SwitchPortGroupRequestMode API is called when this action is executed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMEthTrcvMode [ECUC_BswM_01070] | | |
| **Parent Container** | BswMEthIfSwitchPortGroupRequestMode | | |
| **Description** | This parameter contains the mode which will be requested. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_ETH_MODE_ACTIVE | enable the port group | |
| | BSWM_ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST | Enable the port group and request to trigger a wake-up on the network. | |
| | BSWM_ETH_MODE_DOWN | disable the port group | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **Name** | BswMEthIfSwitchPortGroupRef [ECUC_BswM_01069] |
| **Parent Container** | BswMEthIfSwitchPortGroupRequestMode |
| **Description** | This is a reference to the Ethernet Interface Switch Port Group which will receive the request. |
| **Multiplicity** | 1 |
| **Type** | Symbolic name reference to EthIfSwitchPortGroup |
| **Post-Build Variant Value** | true |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.66 BswMFrSMAllSlots

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01037] |
| **Container Name** | BswMFrSMAllSlots |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameter(s) for the action to request an exit from Flexray KeySlotOnlyMode. FrSM_AllSlots is called when this action is executed. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMFrSMAllSlotsNetworkHandleRef [ECUC_BswM_01038] | | |
| **Parent Container** | BswMFrSMAllSlots | | |
| **Description** | This references the FlexRay cluster. The reference corresponds to the parameter "NetworkHandle" of the function FrSM_AllSlots. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.67 BswMJ1939DcmStateSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01032] |
| **Container Name** | BswMJ1939DcmStateSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to a switch of the J1939 Diagnostic Communication Managers network state for a J1939 node. J1939Dcm_SetState is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939DcmRequestedState [ECUC_BswM_01035] | | |
| **Parent Container** | BswMJ1939DcmStateSwitch | | |
| **Description** | This parameter describes the communication state of the J1939 Diagnostic Communication Manager and corresponds to the parameter "newState" of the function J1939Dcm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_J1939DCM_STATE_OFFLINE | | |
| | BSWM_J1939DCM_STATE_ONLINE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939DcmChannelRef [ECUC_BswM_01033] | | |
| **Parent Container** | BswMJ1939DcmStateSwitch | | |
| **Description** | This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.<br><br>This reference corresponds to the parameter "channel" of the function J1939Dcm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939DcmNodeRef [ECUC_BswM_01034] | | |
| **Parent Container** | BswMJ1939DcmStateSwitch | | |
| **Description** | This reference points to a J1939NmNode and provides access to the unique J1939NmNodeId.This reference corresponds to the parameter "node" of the function J1939Dcm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to J1939NmNode | | |
| **Post-Build Variant Value** | false | | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.68 BswMJ1939RmStateSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00998] |
| **Container Name** | BswMJ1939RmStateSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to a switch of the J1939 Request Managers network state for a J1939 node.<br>J1939Rm_SetState is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939RmRequestedState [ECUC_BswM_01002] | | |
| **Parent Container** | BswMJ1939RmStateSwitch | | |
| **Description** | This parameter describes the communication state of the J1939 Request Manager and corresponds to the parameter "new state" of the function J1939Rm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_J1939RM_STATE_OFFLINE | | |
| | BSWM_J1939RM_STATE_ONLINE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939RmChannelRef [ECUC_BswM_01000] | | |
| **Parent Container** | BswMJ1939RmStateSwitch | | |
| **Description** | This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.<br><br>This reference corresponds to the parameter "channel" of the function J1939Rm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939RmNodeRef [ECUC_BswM_01001] | | |
| **Parent Container** | BswMJ1939RmStateSwitch | | |
| **Description** | This reference points to a J1939NmNode and provides access to the unique J1939NmNodeId.<br><br>This reference corresponds to the parameter "node" of the function J1939Rm_SetState. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to J1939NmNode | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.69 BswMLinScheduleSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00827] |
| **Container Name** | BswMLinScheduleSwitch |
| **Parent Container** | BswMAvailableActions |

| | |
|---|---|
| **Description** | This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured.<br><br>The configuration for the "network" parameter can be accessed via the reference LinSMComMNetworkHandleRef contained in the parent container LinSMChannel of the container referenced by BswMLinScheduleRef. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMLinScheduleRef [ECUC_BswM_00842] | | |
| **Parent Container** | BswMLinScheduleSwitch | | |
| **Description** | This is a reference to the LIN schedule table that the LIN SM shall change to.<br><br>This reference corresponds to the parameter "schedule" of the function LinSM_ScheduleRequest. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to LinSMSchedule | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.70 BswMNMControl

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00837] |
| **Container Name** | BswMNMControl |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to enabling and disabling of Network Management communication. Disabling of NM communication can be requested by DCM. Nm_EnableCommunication or Nm_DisableCommunication is called when this action is configured. |
| **Configuration Parameters** | |

| | | |
|---|---|---|
| **Name** | BswMNMAction [ECUC_BswM_00838] | |
| **Parent Container** | BswMNMControl | |
| **Description** | This parameter specifies if the communication of the corresponding NM channel should be enabled or disabled. | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BSWM_NM_DISABLE | |

| | | | |
|---|---|---|---|
| **Post-Build Variant Value** | BSWM_NM_ENABLE<br>false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMNetworkHandleRef [ECUC_BswM_00999] | | |
| **Parent Container** | BswMNMControl | | |
| **Description** | This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.<br><br>This reference corresponds to the parameter "NetworkHandle" of the function Nm_EnableCommunication and Nm_DisableCommunication. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComMChannel | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME,<br>VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.71 BswMPduGroupSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00828] |
| **Container Name** | BswMPduGroupSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes references to the PDU groups that shall be enabled and disabled. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMPduGroupSwitchReinit [ECUC_BswM_00913] | | |
| **Parent Container** | BswMPduGroupSwitch | | |
| **Description** | This parameter defines if the data of the I-PDU, the shadow buffers of included signal groups, etc. are reinitialized when a PDU Group is started. This parameter corresponds to the parameter "initialize" of the function Com_IpduGroupStart. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMDisabledPduGroupRef [ECUC_BswM_00850] | | |
| **Parent Container** | BswMPduGroupSwitch | | |
| **Description** | This is a reference to a PDU Group that should be disabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_IpduGroupStop. | | |
| **Multiplicity** | 0..* | | |
| **Type** | Symbolic name reference to ComIPduGroup | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEnabledPduGroupRef [ECUC_BswM_00849] | | |
| **Parent Container** | BswMPduGroupSwitch | | |
| **Description** | This is a reference to a PDU Group that should be enabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_IpduGroupStart. | | |
| **Multiplicity** | 0..* | | |
| **Type** | Symbolic name reference to ComIPduGroup | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

**[SWS_BswM_CONSTR_00004]** ⌈The BswM shall reject configurations where a `BswMPduGroupSwitch` container has a `BswMDisabledPduGroupRef` and a `BswMEnabledPduGroupRef` which reference the same PDU Group.⌋*()*

**[SWS_BswM_00273]** ⌈When a `BswMPduGroupSwitch` action is executed, the BswM shall call `Com_IpduGroupStart` for each `BswMEnabledPduGroupRef`, and call `Com_IpduGroupStop` for each `BswMDisabledPduGroupRef`. The ordering of these calls to Com is undefined.⌋*()*

Note: If a strict ordering of the calls to `Com_IpduGroupStart`, or `Com_IpduGroupStop` is required, then this can be achieved by configuring individual actions (`BswMPduGroupSwitch`, each with just a single BswM*PduGroupRef) within an ordered action list.

### 10.2.72 BswMPduRouterControl

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00853] |
| **Container Name** | BswMPduRouterControl |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to enabling and disabling of routing of Routing Path Groups in the PDU Router. PduR_EnableRouting or PduR_DisableRouting is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMPduRouterAction [ECUC_BswM_00854] | | |
| **Parent Container** | BswMPduRouterControl | | |
| **Description** | This parameter specifies if the routing of the corresponding PDU should be enabled or disabled. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_PDUR_DISABLE | | |
| | BSWM_PDUR_ENABLE | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMPduRouterDisableInitBuffer [ECUC_BswM_01036] | | |
| **Parent Container** | BswMPduRouterControl | | |
| **Description** | When BswPduRouterAction is set to BSWM_PDUR_DISABLE and this parameter is set to true, then the call to PduR_DisableRouting will be invoked with parameter "initialize" set to true, otherwise false. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **Name** | BswMPduRoutingPathGroupRef [ECUC_BswM_00855] |
| **Parent Container** | BswMPduRouterControl |
| **Description** | This is a reference to the PDU Routing Path Group for which the routing in the PDU Router should be enabled or disabled.<br><br>This reference corresponds to the parameter "id" of the function PduR_EnableRouting and PduR_DisableRouting. |
| **Multiplicity** | 1..* |
| **Type** | Symbolic name reference to PduRRoutingPathGroup |
| **Post-Build Variant Multiplicity** | true |

| | | | |
|---|---|---|---|
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.73 BswMRteModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01021] |
| **Container Name** | BswMRteModeRequest |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines a mode request that the BswM may send to a SW-C which is acting as a mode-manager. RTE_Write is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMRequestedModeRef [ECUC_BswM_01024] | | |
| **Parent Container** | BswMRteModeRequest | | |
| **Description** | This is a foreign reference to the Mode Declaration used for the mode request | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to MODE-DECLARATION | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMRteModeRequestPortRef [ECUC_BswM_01023] | | |
| **Parent Container** | BswMRteModeRequest | | |
| **Description** | This is a reference to a BswMRteModeRequestPort. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMRteModeRequestPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.74 BswMRteStart

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01073] |
| **Container Name** | BswMRteStart |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines the action to start the Rte from BswM. |
| **Configuration Parameters** | |

| **No Included Containers** |
|---|

**[SWS_BswM_00271]** ⌈If the action `BswMRteStart` is configured, the function `Rte_Start(void)` shall be called by the BswM when the action is executed.⌋*(SRS_ModeMgm_09178)*

### 10.2.75 BswMRteStop

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01074] |
| **Container Name** | BswMRteStop |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines the action to stop the Rte from BswM |
| **Configuration Parameters** | |

| **No Included Containers** |
|---|

**[SWS_BswM_00272]** ⌈If the action `BswMRteStop` is configured, the function `Rte_Stop(void)` shall be called by the BswM when the action is executed.⌋*(SRS_ModeMgm_09178)*

### 10.2.76 BswMRteSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00803] |
| **Container Name** | BswMRteSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container defines a mode switch indication that the BswM provides to the SW-C that need to be notified about the mode switch. RTE_Switch is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMRteSwitchPortRef [ECUC_BswM_00952] | | |
| **Parent Container** | BswMRteSwitch | | |
| **Description** | This is a reference to the BswMSwitchPort. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMSwitchPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSwitchedMode [ECUC_BswM_00896] | | |
| **Parent Container** | BswMRteSwitch | | |
| **Description** | This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to MODE-DECLARATION | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.77 BswMSchMSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00899] |
| **Container Name** | BswMSchMSwitch |
| **Parent Container** | BswMAvailableActions |

| | |
|---|---|
| **Description** | This container defines a mode switch indication that the BswM provides to the SW-C that need to be notified about the mode switch. SchM_Switch is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSchMSwitchedMode [ECUC_BswM_00901] | | |
| **Parent Container** | BswMSchMSwitch | | |
| **Description** | This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group. | | |
| **Multiplicity** | 1 | | |
| **Type** | Foreign reference to MODE-DECLARATION | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSchMSwitchPortRef [ECUC_BswM_01080] | | |
| **Parent Container** | BswMSchMSwitch | | |
| **Description** | This is a reference to the BswMSwitchPort. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMSwitchPort | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.78 BswMSdClientServiceModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00974] |
| **Container Name** | BswMSdClientServiceModeRequest |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to the selection of an client service of Sd. Sd_ClientServiceSetState is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdClientServiceState [ECUC_BswM_01016] | | |
| **Parent Container** | BswMSdClientServiceModeRequest | | |
| **Description** | This parameter specifies if the corresponding client service shall be released or requested. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_SD_CLIENT_SERVICE_RELEASED | Client service shall be released | |
| | BSWM_SD_CLIENT_SERVICE_REQUESTED | Client service shall be requested | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdClientMethodsRef [ECUC_BswM_01009] | | |
| **Parent Container** | BswMSdClientServiceModeRequest | | |
| **Description** | This is a reference to a client service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdClientService | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.79 BswMSdConsumedEventGroupModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01004] |
| **Container Name** | BswMSdConsumedEventGroupModeRequest |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to the selection of a consumed EventGroup of Sd. Sd_ConsumedEventGroupSetState is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdConsumedEventGroupState [ECUC_BswM_01017] | | |
| **Parent Container** | BswMSdConsumedEventGroupModeRequest | | |
| **Description** | This parameter specifies if the corresponding consumed event group shall be released or requested. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_SD_CONSUMED_EVENTGROUP_RELEASED | Event group shall be released. | |
| | BSWM_SD_CONSUMED_EVENTGROUP_REQUESTED | Event group shall be requested. | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdConsumedEventGroupRef [ECUC_BswM_01010] | | |
| **Parent Container** | BswMSdConsumedEventGroupModeRequest | | |
| **Description** | This is a reference to an eventGroup that is defined within a client service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdConsumedEventGroup | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| |
|---|
| **No Included Containers** |

### 10.2.80 BswMSdServerServiceModeRequest

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01005] |
| **Container Name** | BswMSdServerServiceModeRequest |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to the selection of a server service of Sd. Sd_ServerServiceSetState is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdServerServiceState [ECUC_BswM_01015] | | |
| **Parent Container** | BswMSdServerServiceModeRequest | | |
| **Description** | This parameter specifies if the corresponding server service shall be down or available. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_SD_SERVER_SERVICE_AVAILABLE | Server service shall be available. | |
| | BSWM_SD_SERVER_SERVICE_DOWN | Server service shall be down. | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdServerMethodsRef [ECUC_BswM_01007] | | |
| **Parent Container** | BswMSdServerServiceModeRequest | | |
| **Description** | This is a reference to a server service in the Sd module. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to SdServerService | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.81 BswMSdServiceGroupSwitch

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01077] |
| **Container Name** | BswMSdServiceGroupSwitch |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes references to the SdServiceGroups that shall be enabled and disabled. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMDisabledSdServiceGroupRef [ECUC_BswM_01078] | | |
| **Parent Container** | BswMSdServiceGroupSwitch | | |
| **Description** | This is a reference to a SdServiceGroup that should be disabled. This reference corresponds to the parameter "ServiceGroupId" of the function Sd_ServiceGroupStop. | | |
| **Multiplicity** | 0..* | | |
| **Type** | Symbolic name reference to SdServiceGroup | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEnabledSdServiceGroupRef [ECUC_BswM_01079] | | |
| **Parent Container** | BswMSdServiceGroupSwitch | | |
| **Description** | This is a reference to a SdServiceGroup that should be enabled. This reference corresponds to the parameter "ServiceGroupId" of the function Sd_ServiceGroupStart. | | |
| **Multiplicity** | 0..* | | |
| **Type** | Symbolic name reference to SdServiceGroup | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

**[SWS_BswM_CONSTR_00005]** ⌈The BswM shall reject configurations where a `BswMSdServiceGroupSwitch` container has a `BswMDisabledSdService-GroupRef` and a `BswMEnabledSdServiceGroupRef` which reference the same Sd-ServiceGroup.⌋*()*

**[SWS_BswM_00280]** ⌈When a BswMSdServiceGroupSwitch action is executed, the BswM shall call Sd_ServiceGroupStart for each BswMEnabledSdServiceGroupRef, and call Sd_ServiceGroupStop for each BswMDisabledSdServiceGroupRef. The ordering of these calls to Sd is undefined.

Note: If a strict ordering of the calls to Sd_ServiceGroupStart or Sd_ServiceGroupStop is required, then this can be achieved by configuring individual actions (BswMSdServiceGroupSwitch, each with just a single BswM*ServiceGroupRef) within an ordered action list.⌋*()*

### 10.2.82 BswMSwitchIPduMode

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00958] |
| **Container Name** | BswMSwitchIPduMode |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to the selection of the transmission mode an I-PDU to be sent by COM.<br>Com_SwitchIpduTxMode is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMSwitchIPduModeValue [ECUC_BswM_00960] | | |
| **Parent Container** | BswMSwitchIPduMode | | |
| **Description** | This parameter defines which transmission mode shall be selected during this call.<br><br>This parameter corresponds to the parameter "Mode" of the function Com_SwitchIpduTxMode. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSwitchIPduModeRef [ECUC_BswM_00959] | | |
| **Parent Container** | BswMSwitchIPduMode | | |
| **Description** | This is a reference to an I-PDU for which the transmission mode shall be set.<br><br>This reference corresponds to the parameter "PduId" of the function Com_SwitchIpduTxMode. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to ComIPdu | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

### 10.2.83 BswMTimerControl

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_01059] |
| **Container Name** | BswMTimerControl |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters for the action to start or to stop a timer. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMTimerAction [ECUC_BswM_01060] | | |
| **Parent Container** | BswMTimerControl | | |
| **Description** | Specify the action for the timer. The timer can be started or stopped. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | BSWM_TIMER_START | | |
| | BSWM_TIMER_STOP | | |
| **Default Value** | BSWM_TIMER_START | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMTimerValue [ECUC_BswM_01061] | | |
| **Parent Container** | BswMTimerControl | | |
| **Description** | Specify the timer value (in seconds) that is used when the timer is started. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | ]0 .. INF[ | | |
| **Default Value** | | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMTimerRef [ECUC_BswM_01062] | | |
| **Parent Container** | BswMTimerControl | | |
| **Description** | Specify the Timer for which the timer action shall be executed. | | |
| **Multiplicity** | 1 | | |
| **Type** | Reference to BswMTimer | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| |
|---|
| **No Included Containers** |

### 10.2.84 BswMTriggerIPduSend

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00906] |
| **Container Name** | BswMTriggerIPduSend |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all parameters related to the triggering of an I-PDU to be sent by COM. Com_TriggerIPDUSend is called when this action is configured. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMTriggeredIPduRef [ECUC_BswM_00907] | | |
| **Parent Container** | BswMTriggerIPduSend | | |
| **Description** | This is a reference to an I-PDU that should be triggered for transmission.<br><br>This reference corresponds to the parameter "PduId" of the function Com_TriggerIPDUSend. | | |
| **Multiplicity** | 1..* | | |
| **Type** | Symbolic name reference to ComIPdu | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.85 BswMUserCallout

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00834] |
| **Container Name** | BswMUserCallout |
| **Parent Container** | BswMAvailableActions |
| **Description** | This container includes all details needed for a user defined function call. |
| **Configuration Parameters** | |

| | |
|---|---|
| **Name** | BswMUserCalloutFunction [ECUC_BswM_00843] |
| **Parent Container** | BswMUserCallout |
| **Description** | This parameter specifies the complete function call including all parameters. The parameters are specified during configuration time, and cannot be changed during run time. Any return values passed by the callout will be ignored.<br><br>Example usage can be: Actions to initialize other BSW modules Action to call NvM_ReadAll() Action to call NvM_WriteAll() |
| **Multiplicity** | 1 |
| **Type** | EcucStringParamDef |
| **Default Value** | |
| **Regular Expression** | |
| **Post-Build Variant Value** | false |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.86 BswMActionList

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00809] | | |
| **Container Name** | BswMActionList | | |
| **Parent Container** | BswMModeControl | | |
| **Description** | Each instance of this container defines an action list that is invoked based on the BswM Rules. An action list contains a list of numbered action items to be processed. An action list can also include other action lists. | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Configuration Parameters** | | | |

| | | |
|---|---|---|
| **Name** | BswMActionListExecution [ECUC_BswM_00894] | |
| **Parent Container** | BswMActionList | |
| **Description** | This parameter controls if the corresponding action list shall be executed every time the rule is evaluated or only when the result of the evaluation changes. This parameter does not have an effect when this action list is executed within another action list. | |
| **Multiplicity** | 1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | BSWM_CONDITION | Action list shall be executed every time the rule is evaluated. |
| | BSWM_TRIGGER | Action list shall be executed every time the result of the evaluation changes. |
| **Post-Build Variant Value** | false | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMActionListPriority [ECUC_BswM_01076] | | |
| **Parent Container** | BswMActionList | | |
| **Description** | This controls the order of execution, in the case when multiple action lists are executed during a single mode control cycle. Highest priority action list will be executed first. Zero (0) is lowest priority, and 255 is highest priority. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 255 | | |
| **Default Value** | 0 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMActionListItem | 1..* | This container defines an item in an action list. |

### 10.2.87 BswMActionListItem

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00823] | | |
| **Container Name** | BswMActionListItem | | |
| **Parent Container** | BswMActionList | | |
| **Description** | This container defines an item in an action list. | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Configuration Parameters** | | | |

| | |
|---|---|
| **Name** | BswMAbortOnFail [ECUC_BswM_00902] |
| **Parent Container** | BswMActionListItem |
| **Description** | This parameter defines if the execution of the action list shall be aborted if this specific action returns E_NOT_OK. Note that this is only applicable for actions that have E_NOT_OK as a possible return value. |
| **Multiplicity** | 1 |
| **Type** | EcucBooleanParamDef |
| **Default Value** | false |
| **Post-Build Variant Value** | true |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMActionListItemIndex [ECUC_BswM_00824] | | |
| **Parent Container** | BswMActionListItem | | |
| **Description** | This parameter defines the index of the action in the action list. It is used define in which order the actions shall be performed. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 255 | | |
| **Default Value** | | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMReportFailRuntimeErrorId [ECUC_BswM_01050] | | |
| **Parent Container** | BswMActionListItem | | |
| **Description** | If this parameter is configured, and this specific action returns E_NOT_OK, the BswM will report a Det Runtime Error. The ErrorId reported in the Runtime Error is given by the value configured in this parameter. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 128 .. 255 | | |
| **Default Value** | 128 | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Post-Build Variant Value** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMActionListItemRef [ECUC_BswM_00825] | | |
| **Parent Container** | BswMActionListItem | | |
| **Description** | The action item can either be an atomic action or a reference to another action list or rule. | | |
| **Multiplicity** | 1 | | |
| **Type** | Choice reference to [BswMAction, BswMActionList, BswMRule] | | |
| **Post-Build Variant Value** | true | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME |
| | **Post-build time** | X | VARIANT-POST-BUILD |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

### 10.2.88 BswMRteModeRequestPort

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_01022] | | |
| **Container Name** | BswMRteModeRequestPort | | |
| **Parent Container** | BswMModeControl | | |
| **Description** | This container defines a mode request port which the BswM may utilize to send a mode request to a SW-C which is acting as a mode-manager. If this container is referenced by a BswMRteModeRequest, the BswM shall create a corresponding PPort in its service description. | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Configuration Parameters** | | | |

| | | | |
|---|---|---|---|
| **Name** | BswMRteModeRequestPortInterfaceRef [ECUC_BswM_01025] | | |
| **Parent Container** | BswMRteModeRequestPort | | |
| **Description** | This is an instance reference to the variable data prototype used for the mode request. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Instance reference to VARIABLE-DATA-PROTOTYPE context: SW-C OMPONENT-PROTOTYPE*PORT-PROTOTYPE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |

| | |
|---|---|
| **Scope / Dependency** | scope: local<br>dependency: BswMRteModeRequestVariableDataPrototypeSRRef == NULL |

| | | | |
|---|---|---|---|
| **Name** | BswMRteModeRequestVariableDataPrototypeSRRef [ECUC_BswM_01057] | | |
| **Parent Container** | BswMRteModeRequestPort | | |
| **Description** | This is a foreign reference to a VariableDataPrototype used for the mode request. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Foreign reference to VARIABLE-DATA-PROTOTYPE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local<br>dependency: BswMRteModeRequestPortInterfaceRef == NULL | | |

| |
|---|
| **No Included Containers** |

### 10.2.89 BswMSwitchPort

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_BswM_00950] | | |
| **Container Name** | BswMSwitchPort | | |
| **Parent Container** | BswMModeControl | | |
| **Description** | Represents an output mode-switch port to be generated by the BswM. If BswMModeSwitchInterfaceRef is configured then a PPortPrototype is generated in the SWCD. If BswMSchMModeDeclarationGroupRef is configured then a ModeDeclarationGroupPrototype is generated in the ProvidedModeGroups of the BSWMD. If both BswMModeSwitchInterfaceRef and BswMSchMModeDeclarationGroupRef are configured then an SwcBswSynchronizedModeGroupPrototype is also generated in the BSWMD (see Chapter 6.11 of the BSW Module Description Template SWS and EXP ModemanagementGuide).. | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE, VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Configuration Parameters** | | | |

| | | | |
|---|---|---|---|
| **Name** | BswMModeSwitchInterfaceRef [ECUC_BswM_00951] | | |
| **Parent Container** | BswMSwitchPort | | |
| **Description** | Reference to the ModeSwitchInterface from which the BswM will generate a PPortPrototype. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Foreign reference to MODE-SWITCH-INTERFACE | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSchMModeDeclarationGroupRef [ECUC_BswM_01031] | | |
| **Parent Container** | BswMSwitchPort | | |
| **Description** | Reference to the ModeDeclarationGroup from which the BswM will generate a ModeDeclarationGroupPrototype. | | |
| **Multiplicity** | 0..1 | | |
| **Type** | Foreign reference to MODE-DECLARATION-GROUP | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **No Included Containers** |
|---|

**[SWS_BswM_CONSTR_00006]** ⌈If the BswMSwitchPort is referred by any BswMRteSwitch actions or BswMModeSwitchErrorEvents then BswMModeSwitchInterfaceRef shall be configured. If the BswMSwitchPort is referred by any BswMSchMSwitch actions then BswMSchMModeDeclarationGroupRef shall be configured. At least one of the contained references shall be configured.⌋*()*

### 10.2.90 BswMGeneral

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00800] |
| **Container Name** | BswMGeneral |
| **Parent Container** | BswM |
| **Description** | General configuration parameters of the Basic SW Mode Manager. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMCanSMEnabled [ECUC_BswM_00938] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable CanSM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMComMEnabled [ECUC_BswM_00939] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable ComM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMDcmEnabled [ECUC_BswM_00940] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable Dcm module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMDevErrorDetect [ECUC_BswM_00811] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | Switches the development error detection and notification on or off.<br>• true: detection and notification is enabled.<br>• false: detection and notification is disabled. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEcuMEnabled [ECUC_BswM_00941] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable EcuM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEthIfEnabled [ECUC_BswM_01072] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable EthIf module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMEthSMEnabled [ECUC_BswM_00942] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable EthSM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMFrSMEnabled [ECUC_BswM_00943] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable FrSM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMGenericRequestEnabled [ECUC_BswM_00949] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable Generic Request related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939DcmEnabled [ECUC_BswM_00987] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | Enable/disable J1939Dcm module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMJ1939NmEnabled [ECUC_BswM_00965] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | Enable/disable J1939Nm module related BswM API. true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMLinSMEnabled [ECUC_BswM_00944] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable LinSM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMLinTPEnabled [ECUC_BswM_00945] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable LinTP module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMMainFunctionPeriod [ECUC_BswM_00813] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | The cycle time of the periodic main function of BswM. Defined in seconds . | | |
| **Multiplicity** | 0..1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | ]0 .. INF[ | | |
| **Default Value** | | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | VARIANT-PRE-COMPILE |
| | **Link time** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **Name** | BswMNmEnabled [ECUC_BswM_01071] |
| **Parent Container** | BswMGeneral |
| **Description** | enable/disable Nm module related BswM API: true: Enabled false: Disabled |
| **Multiplicity** | 1 |
| **Type** | EcucBooleanParamDef |
| **Default Value** | false |
| **Post-Build Variant Value** | false |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMNvMEnabled [ECUC_BswM_00946] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable NvM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSchMEnabled [ECUC_BswM_00947] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable SchM module related BswM API: true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMSdEnabled [ECUC_BswM_01047] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable Sd module related BswM API. true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMVersionInfoApi [ECUC_BswM_00812] | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | Switches the possibility to read the version information with the service BswM_GetVersionInfo(). true: Enabled false: Disabled | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **Name** | BswMWdgMEnabled [ECUC_BswM_00948] (Obsolete) | | |
| **Parent Container** | BswMGeneral | | |
| **Description** | enable/disable WdgM module related BswM API: true: Enabled false: Disabled<br><br>**Tags:**<br>atp.Status=obsolete | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default Value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| BswMUserIncludeFiles | 0..1 | Collection of header file names which shall be included by the BswM. |

**[SWS_BswM_00279]** ⌈The availability of private header files (e.g. BswM_ComM.h) between the BswM and other basic software modules depend on the configuration parameters (BswM<BSWModule>Enabled). If the configuration parameter is set to

false, then no private header shall be provided by BswM for that BSWModule.⌋ *(SRS_BSW_00301)*

### 10.2.91 BswMUserIncludeFiles

| | |
|---|---|
| **SWS Item** | [ECUC_BswM_00954] |
| **Container Name** | BswMUserIncludeFiles |
| **Parent Container** | BswMGeneral |
| **Description** | Collection of header file names which shall be included by the BswM. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **Name** | BswMUserIncludeFile [ECUC_BswM_00955] | | |
| **Parent Container** | BswMUserIncludeFiles | | |
| **Description** | Header file name which shall be included by the BswM. The value of this parameter shall be used as h-char-sequence or q-char-sequence according to ISO C99. The parameter value shall not represent a path. | | |
| **Multiplicity** | 1..* | | |
| **Type** | EcucStringParamDef | | |
| **Default Value** | | | |
| **Regular Expression** | | | |
| **Post-Build Variant Multiplicity** | false | | |
| **Post-Build Variant Value** | false | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | – | |
| | **Post-build time** | – | |
| **Scope / Dependency** | scope: local | | |

**No Included Containers**

## 10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in SWS_BSWGeneral.

# A Not applicable requirements

**[SWS_BswM_09999]** ⌈These requirements are not applicable to this specification.⌋*(SRS_BSW_00405, SRS_BSW_00170, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00336, SRS_BSW_00339, SRS_BSW_00409)*