Document Title	Specification of Basic Software Mode Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	313
Document Classification	Standard

Document Version	1.2.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History

Date	Version	Changed by	Change Description
09.12.2011	1.2.0	AUTOSAR Administration	• Support of Mode Machine Instances assigned to the SchM • Include of user defined header files • Possibility to provide an initial value for a BswMModeRequestPort
22.10.2010	1.1.0	AUTOSAR Administration	• Include file BswMUserCallout.h added. This user defined header file contains declarations of the call out functions. • Requirement that the BswM module shall perform inter module version checks added • Information added for each configurable action which API to call • Functions BswM_TriggerSlaveRTEStop and BswM_TriggerStartUpPhase2 added to control the start and stop of the RTE on slave cores
30.11.2009	1.0.0	AUTOSAR Administration	Initial release

Disclaimer

Advice for users

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

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

Table of Contents

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:*
AMM	Application Mode Management
BSW	Basic Software
BswM	BSW Mode Manager
Dem	Diagnostic Event Manager
Det	Development Error Tracer
ECU	Electric Control Unit
RTE	Real Time Environment
VMM	Vehicle Mode Management

Table 1: Table of acronyms and abbreviations

3 Related documentation

3.1 Input documents

[1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf.pdf

[3] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[4] Requirements on Mode Management
AUTOSAR_SRS_ModeManagement.pdf

[5] Specification of Communication
AUTOSAR_SWS_COM.pdf

[6] Specification of FlexRay State Manager
AUTOSAR_SWS_FlexRayStateManager.pdf

[7] Specification of PDU Router
AUTOSAR_SWS_PDURouter.pdf

[8] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[9] Specification of Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[10] Specification of RTE Software
AUTOSAR_SWS_RTE.pdf

[11] Specification of Diagnostic Communication Manager
AUTOSAR_SWS_DiagnosticCommunicationManager.pdf

[12] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

[13] Specification of LIN State Manager
AUTOSAR_SWS_LINStateManager.pdf

[14] Specification of CAN State Manager
AUTOSAR_SWS_CANStateManager.pdf

[15] Specification of Generic Network Management Interface
AUTOSAR_SWS_NetworkManagementInterface.pdf

[16] Specification of Communication Manager
AUTOSAR_SWS_COMManager.pdf

[17] Specification of Ethernet State Manager
AUTOSAR_SWS_EthernetStateManager.pdf

3.2 Related standards and norms

None.

4 Constraints and assumptions

4.1 Limitations

Maximum one instance of the BSW Mode Manager may be used within an AUTOSAR ECU.

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.

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 - Fixed

When EcuM – Fixed is used it will indicate the current ECU state to the BswM.

5.3 EcuM - Flex

When EcuM – Flex is used BswM will change the EcuM state using standard RTE switch calls to the RTE. EcuM Flex can also indicate the state of its wakeup sources to BswM.

5.4 WdgM

Mode Switch Indications originating from the WdgM go through the BswM for further propagation to the SW-Cs. The WdgM also request reset of partitions via the BswM

5.5 ComM

Mode Switch Indications originating from the ComM go through the BswM for further propagation to the SW-Cs.
The BswM can act as a ComM user requesting communication modes.

5.6 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 enable and disable of deadline monitoring of signals in COM.
BswM can also trigger transmission of an I-PDU.

5.7 PduR

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

5.8 CanSM

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

5.9 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.

5.10 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.

5.11 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.

5.12 EthSM

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

5.13 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 PDU:s.

5.14 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.

5.15 NvM

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

5.16 OS

The features of OS that BswM needs to use is implementation specific.

5.17 File structure

Figure 1: File structure of BSW Mode ManagerFigure 1 shows the file structure of the BSW Mode Manager and the header files of other modules it needs to include.

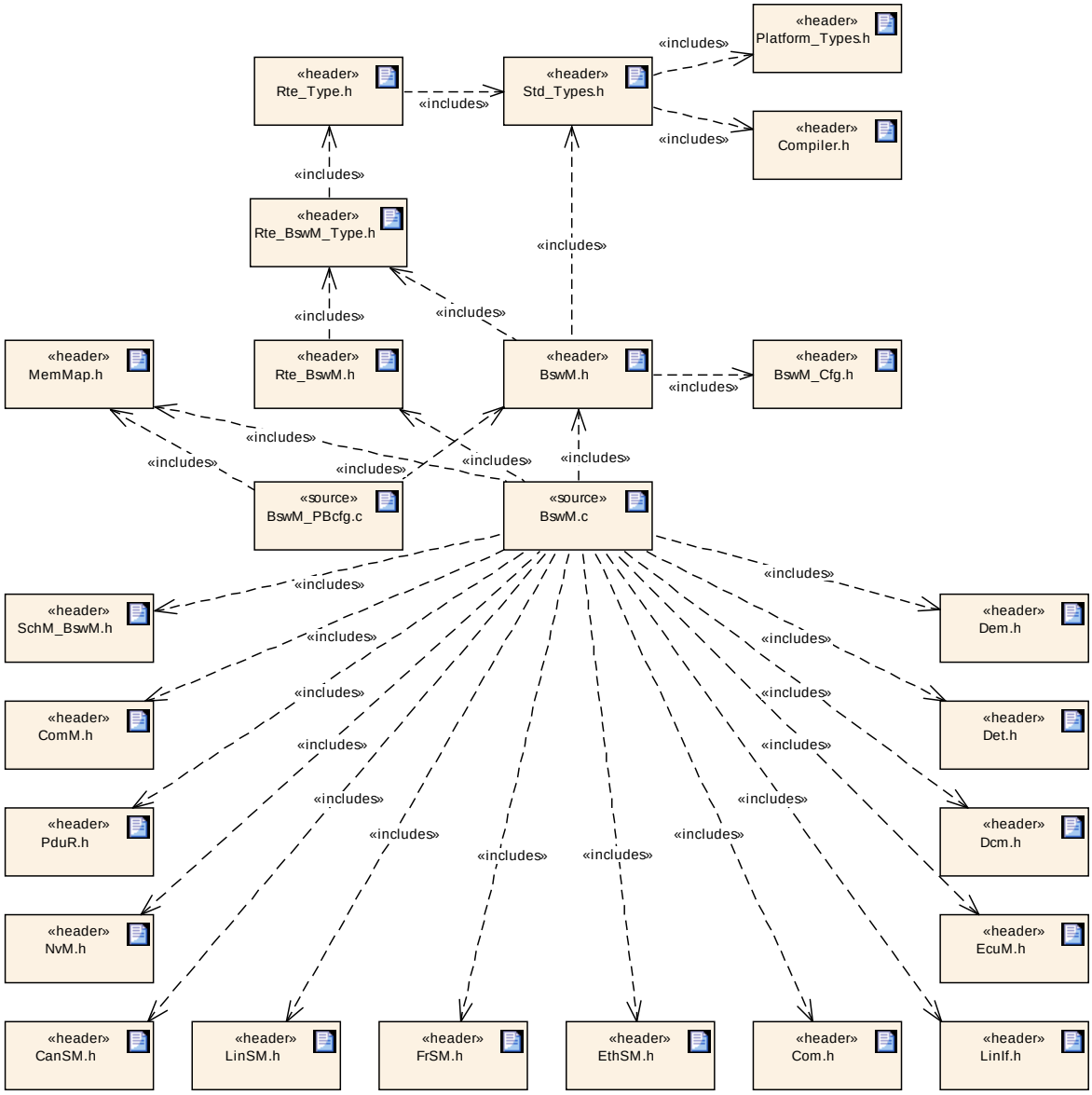


Figure 1: File structure of BSW Mode Manager

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

5.17.1 Code file structure

[BswM0024] ⌈
The code file structure shall not be defined within this specification completely. At this point it shall be pointed out that the code-file structure shall include the following files named:

- BswM_Lcfg.c – for link time configurable parameters and
- BswM_PBcfg.c – for post build time configurable parameters.

These files shall contain all link time and post-build time configurable parameters.⌋ (BSW00380, BSW00419)

5.17.2 Header file structure

[BswM0025] ⌈
The BswM shall include the header files of all other BSW modules which API functions it uses.
Specifically it shall include StdTypes.h and ComStack_Types.h to avoid redefinition of types.

BswM0026:
The BswM module shall provide the following set of header files for other BSW modules to include:

1. BswM header file: BswM.h, BswM_CanSM.h, BswM_LinSM.h, BswM_FrSM.h, BswM_EthSM.h, BswM_RTE.h, BswM_EcuM.h, BswM_COM.h, BswM_ComM.h, BswM_WdgM.h, BswM_DCM.h, User_Callout.H
2. BswM configuration file: BswM_Cfg.h ⌋ (BSW00381, BSW00412, BSW00415)

[BswM0027] ⌈
The module shall include the Dem.h file. By this inclusion the APIs to report errors as well as the required Event Id symbols are included. This specification defines the name of the Event Id symbols which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem_IntErrId.h.⌋()

6 Requirements traceability

Requirement	Description	Satisfied by
BSW00336	These requirements are not applicable to this specification.	BswM9999
BSW00339	These requirements are not applicable to this specification.	BswM9999
BSW00380		BswM0024
BSW00381		BswM0025
BSW00387	These requirements are not applicable to this specification.	BswM9999
BSW00399	These requirements are not applicable to this specification.	BswM9999
BSW00400	These requirements are not applicable to this specification.	BswM9999
BSW00405	These requirements are not applicable to this specification.	BswM9999
BSW00406		BswM0077, BswM0078, BswM0149, BswM0079, BswM0080, BswM0081, BswM0082, BswM0083, BswM0084, BswM0132, BswM0109, BswM0153, BswM0086, BswM0112, BswM0134
BSW00407		BswM0003
BSW00409	These requirements are not applicable to this specification.	BswM9999
BSW00412		BswM0025
BSW00415		BswM0025
BSW00419		BswM0024
BSW009176		BswM0038
BSW09174		BswM0038
BSW09175		BswM0038
BSW09177		BswM0010, BswM0012, BswM0016, BswM0015
BSW09179		BswM0046
BSW09180		BswM0009, BswM0013, BswM0014
BSW09182		BswM0038
BSW09183		BswM0038
BSW09184		BswM0038
BSW09228		BswM0046, BswM0047, BswM0148, BswM0048, BswM0049, BswM0050, BswM0051, BswM0052
BSW09229		BswM0039, BswM0040

BSW09230		BswM0011, BswM0023
BSW101		BswM0002
BSW170	These requirements are not applicable to this specification.	BswM9999

Document: AUTOSAR requirements on Basic Software, general [3].

Requirement	***Satisfied by***
[BSW00344] Reference to link--time configuration	[BswM0021
[BSW00404] Reference to post build time configuration	Chapter 5
[BSW00405] Reference to multiple configuration sets	Not applicable
[BSW00345] Configuration at Compile time	[BswM0020
[BSW159] Automatic configuration	Chapter 10
[BSW167] Static configuration checking	Chapter 10
[BSW171] Configurability of optional functionality	Chapter 10
[BSW170] Data for reconfiguration of AUTOSAR SW--Components	Not applicable
[BSW00380] Separate C--Files for configuration parameters	[BswM0024
[BSW00419] Separate C--Files for pre--compile time configuration parameters	[BswM0024
[BSW00381] Separate configuration header file for pre--compile time parameters	BswM0026
[BSW00412] Separate H--File for configuration parameters	BswM0026
[BSW00383] List dependencies of configuration files	Chapter 5
[BSW00384] List dependencies to other modules	Chapter 5
[BSW00387] Specify the configuration class of callback function	Not applicable
[BSW00388] Introduce containers	Chapter 10
[BSW00389] Containers shall have names	Chapter 10
[BSW00390] Parameter content shall be unique within the module	Chapter 10
[BSW00391] Parameter shall have unique names	Chapter 10
[BSW00392] Parameters shall have a type	Chapter 10
[BSW00393] Parameters shall have a	Chapter 10

range	
[BSW00394] Specify the scope of the parameters	Chapter 10
[BSW00395] List the required parameters (per parameter)	Chapter 10
[BSW00396] Configuration classes	Chapter 10
[BSW00397] Pre--compile--time parameters	Chapter 10
[BSW00398] Link--time parameters	Chapter 10
[BSW00399] Loadable Post--build time parameters	Not applicable
[BSW00400] Selectable Post--build time parameters	Not applicable
[BSW00438] Post Build Configuration Data Structure	Chapter 10
[BSW00402] Published information	Chapter 10
[BSW101] Initialization interface	[BswM0002
[BSW00406] Check module initialization	[BswM0077, [BswM0078, [BswM0079, [BswM0080, [BswM0081, [BswM0082, [BswM0083, [BswM0084, [BswM0086
[BSW00407] Function to read out published parameters	[BswM0003
[BSW00423] Usage of SW--C template to describe BSW modules with AUTOSAR Interfaces	Chapter 7.7
[BSW00336] Shutdown interface	Not Applicable
[BSW00337] Classification of errors	Chapter 7.3
[BSW00338] Detection and Reporting of development errors	Chapter 7.4
[BSW00369] Do not return development error codes via API	Chapter 8
[BSW00339] Reporting of production relevant errors and exceptions	Not Applicable
[BSW00323] API parameter checking	Chapter 8
[BSW00409] Header files for production code error IDs	Not Applicable
[BSW00385] List possible error notifications	Chapter 7.3
[BSW00386] Configuration for detecting an error	Chapter 7.4
[BSW00415] User dependent include files	BswM0026
[BSW00343] Specification and configuration of time	Chapter 10
[BSW00346] Basic set of module files	Chapter 5
[BSW158] Separation of configuration from implementation	Chapter 5
[BSW00370] Separation of callback interface from API	Chapter 8
[BSW00357] Standard API return type	Chapter 8

[BSW00377] Module specific API return types	Chapter 8
[BSW00371] Do not pass function pointers via API	Chapter 8
[BSW00358] Return type of init() functions	Chapter 8
[BSW00414] Parameter of init function	Chapter 8
[BSW00376] Return type and parameters of main processing functions	Chapter 8
[BSW00359] Return type of callback functions	Chapter 8
[BSW00360] Parameters of callback functions	Chapter 8
[BSW00440] Function prototype for callback functions of AUTOSAR Services	Chapter 7.7
[BSW00374] Module vendor identification	Chapter 10
[BSW00379] Module identification	Chapter 10
[BSW003] Version identification	Chapter 10
[BSW00318] Format of module version numbers	Chapter 10
[BSW00321] Enumeration of module version numbers	Chapter 10

Document: AUTOSAR Requirements on Mode Management [4].

Requirement	***Satisfied by***
[BSW09177] Support of a configurable mode arbitration	BswM0010, BswM0012
[BSW09178] Support of Lists of Mode Dependant Actions	BswM0016, BswM0015
[BSW09175] Support of a configurable set of mode dependent enabled and concomitant disabled IPDU groups	BswM0038
[BSW09176] Support of a configurable set of mode dependent to be enabled IPDU groups	BswM0038
[BSW09183] Support of Mode dependent to be reset Signal Initial Values	BswM0038
[BSW09174] Support of “Disable normal Communication”	BswM0038
[BSW09179] Provision of an Interface to allow Mode Requests of SW C’s	Chapter 7.6 and BswM0046
[BSW09180] Arbitration of Mode Requests	BswM0009, BswM0013, BswM0014
[BSW09182] Local propagation of mode change information	BswM0038
[BSW09184] Mode dependent activation and deactivation of IPDU groups	BswM0038

[BSW09228] Provision of an Interface to allow Mode Requests of BSW Modules	BswM0046, BswM0047, BswM0048, BswM0049, BswM0050, BswM0051, BswM0052
[BSW09229] Mode dependent callout	BswM0039, BswM0040
[BSW09230] All actions shall only be performed on mode change	BswM0011, BswM0023

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 which behavior is completely defined by its configuration.
There may be different ways of implementing this, such as generation of the complete BswM based on the configuration, or as a rule interpreter that parses an encoded configuration in 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.
To know what action lists to execute the BswM is required to detect changes in mode arbitration result 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, NAND. Each term in the expression corresponds to a mode request condition. These conditions verify if a requested or indicated mode is EQUAL or NOT_EQUAL to a certain mode. An example rule with two conditions is shown in Figure 2. The rules, and the set of available logical operations are defined as a part of the ECU configuration described in chapter 10.2.

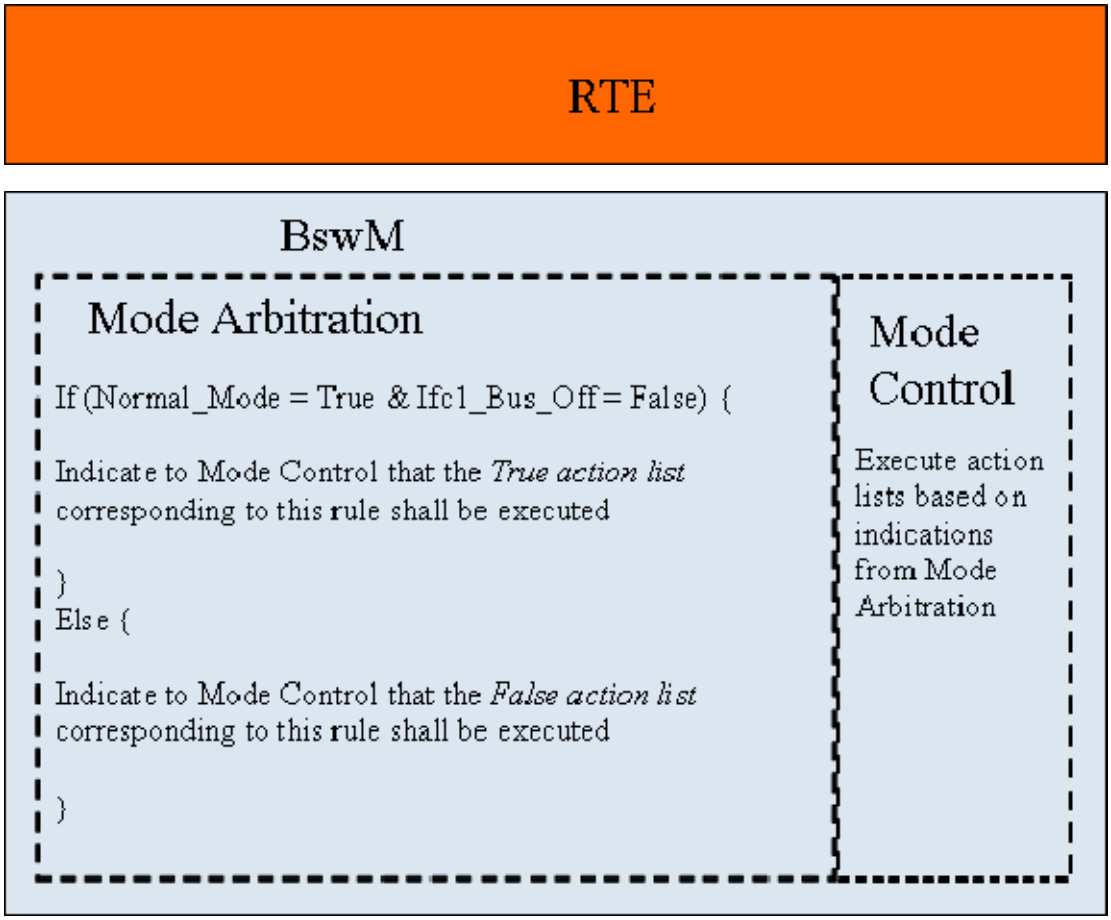


Figure 2: Rule execution in the Mode Arbitration part of the BswM.

7.1.3 Requirements on 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 be produced by other BSW modules such as the DCM. Mode indications are always issued by other BSW modules such as the different bus specific State Managers, EcuM and WdgM. In this document the generic term *mode arbitration request* corresponds either to a mode indication or a mode request.

[BswM0009] 「

The BswM shall perform mode arbitration based on incoming mode requests. 」(BSW09180)

[BswM0035] 「

The BswM shall perform mode arbitration based on incoming mode indications. 」()

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.

[BswM0010] ⌈
The BswM shall perform mode arbitration using predefined rules.⌋(BSW09177)

[BswM0012] ⌈
The mode arbitration rules shall be configurable using the module configuration parameters described in chapter 10.2.⌋(BSW09177)

[BswM0117] ⌈
BswM is not allowed to use results from previous mode arbitrations as input for the logical expressions.⌋()

[BswM0147] ⌈
The mode arbitration rule may be called only in context of an action list.⌋()

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) in the main function of the BswM.

If an action list has the property 'immediate', it is executed in the environment 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 9.1and 9.2.

[BswM0061] ⌈
A mode arbitration rule may contain any combination of immediate and deferred mode arbitration requests.⌋()

[BswM0013] ⌈
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 to BSWM_IMMEDIATE. ⌋ (BSW09180)

[BswM0059] ⌈
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.⌋()

[BswM0014] ⌈
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 to BSWM_DEFERRED.⌋(BSW09180)

[BswM0060] ⌈
All rules that uses at least one deferred mode condition shall be evaluated during every execution of the main function of BswM. ⌋()

[BswM0068] ⌈
BswM shall delay mode arbitration requests received during the processing of its main function until it is finished.⌋()

[BswM0069] ⌈
BswM shall delay mode arbitration requests received during the processing of an *immediate request* until it is finished.⌋()

Note: It is up to the implementer of the BswM to decide how to obtain the behavior required in BswM0068 and BswM0069.

7.1.4 Arbitration Behavior after Initialization

The behavior of the mode arbitration of BswM after initialization is controlled by the configuration parameter BswMModeInitValue. This parameters is configured once for each BswMModeRequestPort in the configuration. After initialization of the BswM mode requests are considered as undefined until the corresponding mode requester have made a mode arbitration request for the first time.

[BswM0064] ⌈
If the parameter BswMModeInitValue does not exist 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.⌋()

[BswM0065] ⌈
BswM rule shall not evaluate a rule as long as any of its mode conditions have an undefined mode/state.⌋()

7.2 Mode Control

The Mode Control part of BswM carries out all actions that should be taken based on 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 BswM can issue a DEM event and/or cancel the currently executing action list.

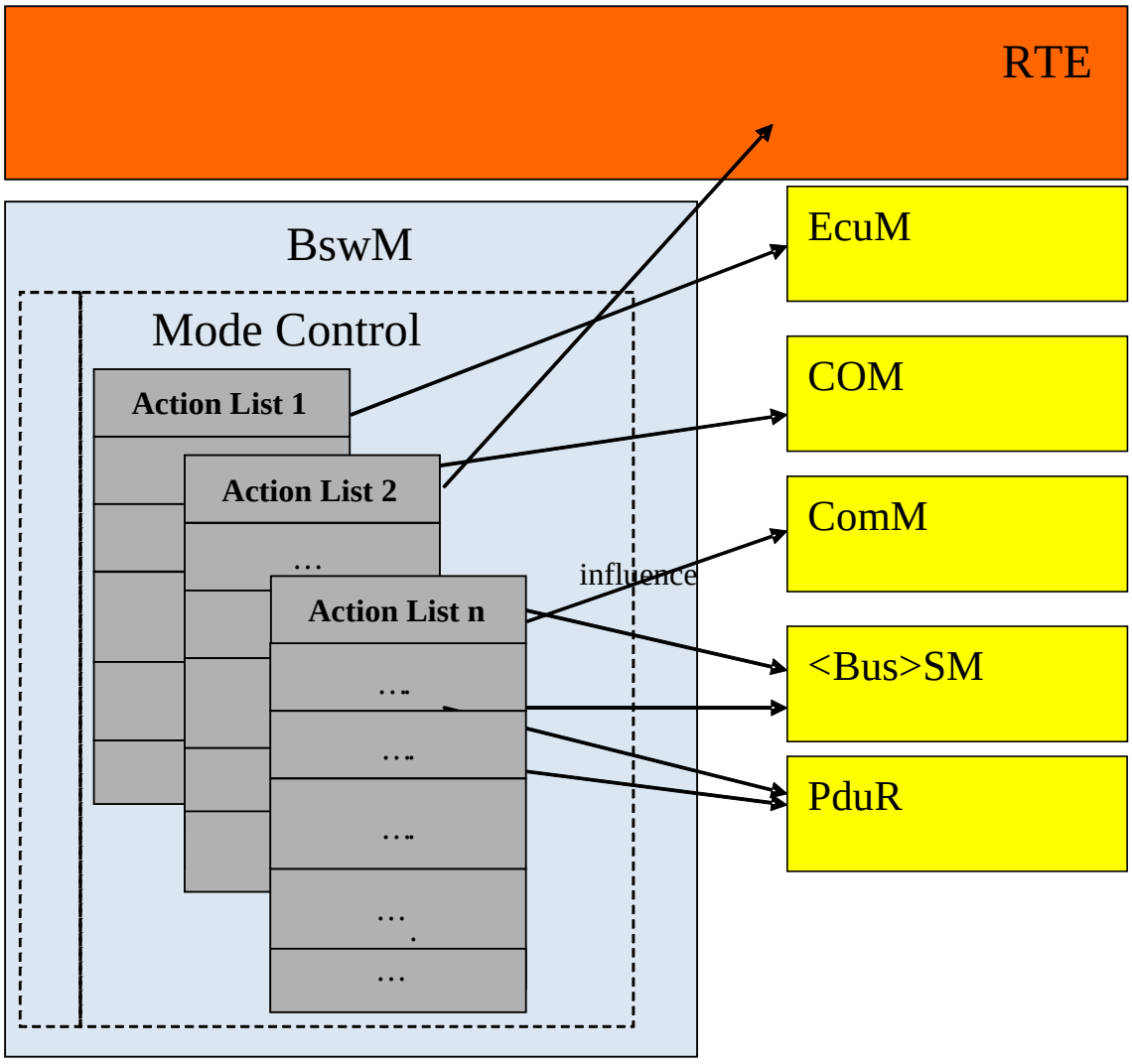


Figure 3: Mode Control in the BswM

As shown in Figure 3, an Action List can hold 1 to N 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 4 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. If the result of a rule evaluation has changed from the previous evaluation, the corresponding Action List is executed.
4. When executing the Action List, the BswM may issue one or several calls to the RTE Switch API [10] 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 originates from a different ECU that is made by a local proxy SW-C.

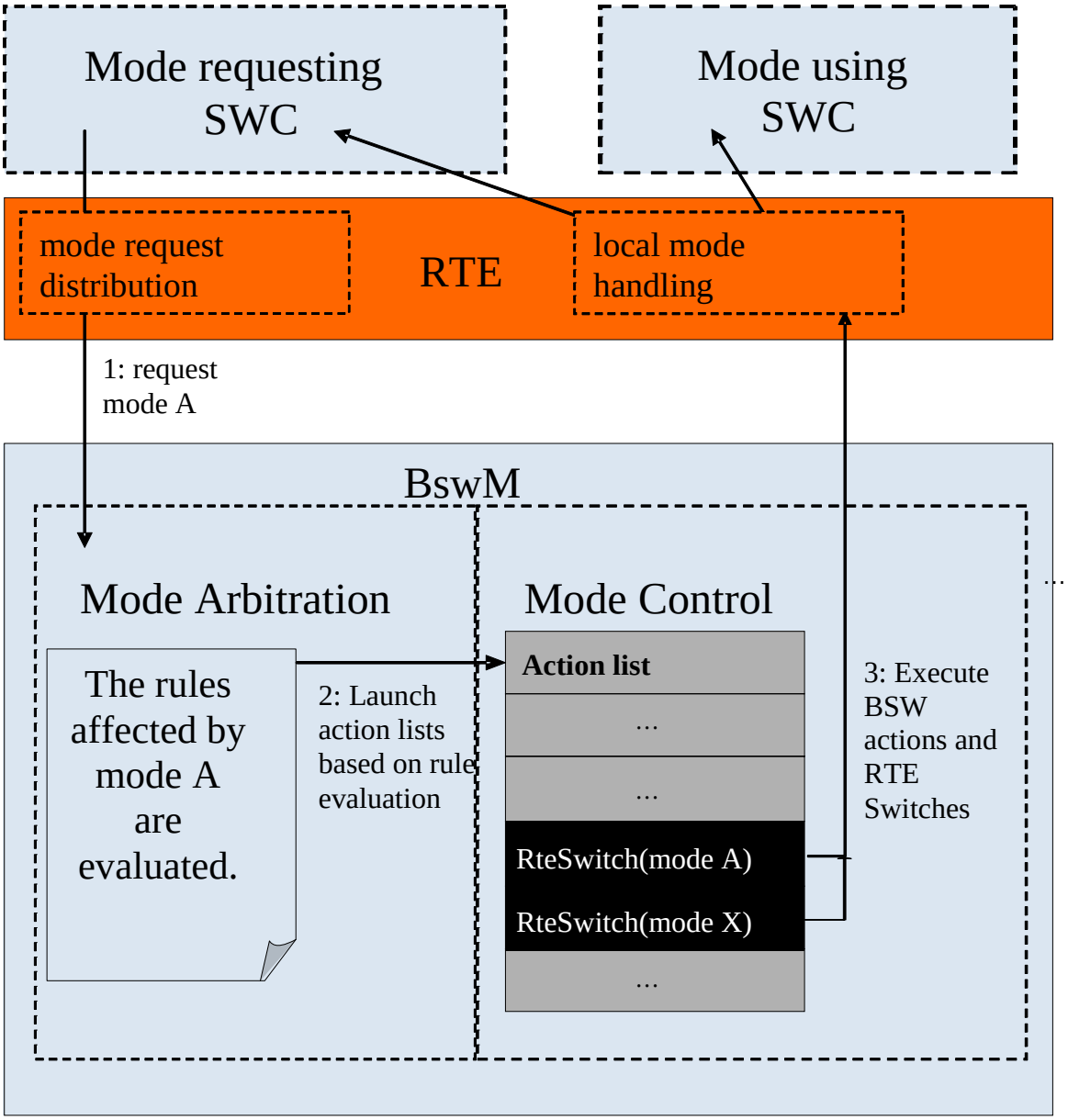


Figure 4: Mode Processing Cycle

7.2.2 Requirements on Mode Control

[BswM0016] ⌈
The BswM shall perform mode control by means of action lists that are executed as a result of rule evaluation in the mode arbitration.⌋(BSW09177)

[BswM0015] ⌈
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. ⌋(BSW09177)

[BswM0017] ⌈
An action list comprises a set of actions that BswM shall execute in an ordered manner. ⌋()

[BswM0018] ⌈
An action list may contain links to other action lists that BswM shall include in the execution.⌋()

[BswM0019] ⌈
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.⌋()

[BswM0067] ⌈If a rule is included in an action list as specified in [BswM0019 any action list execution resulting from that evaluation shall be executed by BswM before it continues to execute the original action list. ⌋()

[BswM0037] ⌈
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 has to be checked by the generator tool.⌋()

[BswM0062] ⌈
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. ⌋()

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.

[BswM0011] ⌈
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.⌋ (BSW09230)

[BswM0023] ⌈

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.」(BSW09230)

[BswM0115]「
If a True action list is configured for conditional execution the BswM shall execute it every time the corresponding rule is evaluated to True.」()

[BswM0116]「
If a False action list is configured for conditional execution the BswM shall execute it every time the corresponding rule is evaluated to False.」()

[BswM0055]「
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”.」()

[BswM0121]「
The BswM shall report a DEM event if an action returns E_NOT_OK and the corresponding BswMReportFailToDem configuration parameter is set to “true”.」()

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.

[BswM0038]「
The following actions shall be available as standard actions available for the BswM by means of selecting corresponding configuration containers and parameters.

- ComM – Setting communication mode of a communication interface
- ComM – Limitation of Communication Mode
- ComM – Call to allow communication in a ComM channel
- LinSM – Setting LIN schedule tables
- FlexRay – Switch to “All Slot Mode”
- COM – Activation/Deactivation of PDU groups with or without resetting of signal initialization values.
- COM – Enabling and disabling of deadline timeout monitoring
- COM – Trigger I-PDU Transmission
- EcuM – Setting ECU Operation Mode
- EcuM – Kill all run requests
- Network Management – Disable and Enable NM Communication
- PduR – Enable/Disable PDU Routing Path Groups, to support mode dependent PDU Gateway.

- Mode switches to the RTE and the BSW Scheduler. 」(BSW09175, BSW009176, BSW09183,BSW09174, BSW09182, BSW09184)

[BswM0039] 「
The BswM shall be able to call any function in the AUTOSAR BSW even though it is not among the standardized actions defined in [BswM0038.」(BSW09229)

[BswM0040] 「
The BswM shall be able to call user defined functions.」(BSW09229)

[BswM0054] 「
The parameters of the user defined functions, and their values, shall be defined at ECU configuration time.」()

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. 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 also affects the action list execution after initialization.

[BswM0066] 「
The BswM shall act according to what is stated in Table 2 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 2: Usage of the BswMRuleInitState configuration parameter

Note: The “true” and “false” action lists are optional for each rule.

7.2.6 Handling of I-PDU group switching

To perform I-PDU group switches (enable/disable) in an efficient and consistent way BswM shall perform the actual I-PDU Group Control function call at the end of processing the main function or an immediate processing request. Basically this means that the I-PDU Group Switch actions manipulates a I-PDU Group Vector that is internal to the BswM and that this vector is passed as a parameter to COM.

[BswM0128] ⌈
BswM shall keep an internal variable of the type Com_IpduGroupVector that is used as an intermediate storage of the results of BswMPduGroupSwitch actions.⌋()

[BswM0129] ⌈
If any BswMPduGroupSwitch action(s) have been performed BswM shall execute the Com_IpduGroupControl command at the end of its processing of the BswM main function or an immediate request processing.⌋()

7.3 Debugging Support

[BswM0070] ⌈
Each variable of the BSW Mode Manager that shall be accessible by AUTOSAR Debugging, shall be defined as global variable.⌋()

[BswM0071] ⌈
All type definitions of variables in the BSW Mode Manager which shall be debugged, shall be accessible by the header file BswM.h.⌋()

[BswM0072] ⌈
The declaration of BSW Mode Manager debugging variables in the header file shall be such, that it is possible to calculate the size of the variables by C-"sizeof".⌋()

[BswM0073] ⌈
Variables in the BSW Mode Manager available for debugging shall be described in the respective Basic Software Module Description.⌋()

7.4 Error classification

[BswM0028] ⌈
Values for production code Event Ids are assigned externally by the configuration of the Dem. They are published in the file Dem_IntErrId.h and included via Dem.h.⌋()

[BswM0029] ⌈
Development error values are of type uint8.⌋()

Type or error	*Relevance*	*Related error code*	*Value [hex]*
A service was called prior to initialization	Development	BSWM_E_NO_INIT	0x01
A null pointer was passed as an argument	Development	BSWM_E_NULL_POINTER	0x02
A parameter was invalid (unspecific)	Development	BSWM_E_INVALID_PAR	0x03
A requesting user was out of range	Development	BSWM_E_REQ_USER_OUT_OF_RANGE	0x04
A requested mode was out of range	Development	BSWM_E_REQ_MODE_OUT_OF_RANGE	0x05
The provided configuration is inconsistent	Development	BSWM_E_PARAM_CONFIG	0x06
A parameter pointer was invalid	Development	BSWM_E_PARAM_POINTER	0x07
An action returned E_NOT_OK	Production	BSWM_E_ACTION_FAILED	

Table 3: Table of development errors

7.5 Error detection

[BswM0030] ⌈
The detection of development errors is configurable (*ON* / *OFF*) at pre-compile time. The switch BSWM_*DEV_ERROR_DETECT* (see chapter 10) shall activate or deactivate the detection of all development errors.⌋()

[BswM0031] ⌈
If the *BSWM_DEV_ERROR_DETECT* switch is enabled API parameter checking is enabled. The detailed description of the detected errors can be found in chapter 8.⌋()

[BswM0032] ⌈
The detection of production code errors cannot be switched off.⌋()

7.6 Error notification

[BswM0033] ⌈
Detected development errors shall be reported to the *Det_ReportError* service of the Development Error Tracer (DET) if the pre-processor switch *BSWM_DEV_ERROR_DETECT* is set (see chapter 10). ⌋()

[BswM0034] ⌈
Production errors shall be reported to Diagnostic Event Manager.⌋()

7.7 Version checking

[BswM0136] ⌈
The BswM module shall perform Inter Module Checks to avoid integration of incompatible files.
The imported included files shall be checked by preprocessing directives. ⌋()

The following version numbers shall be verified:
- <MODULENAME>_AR_RELEASE_MAJOR_VERSION
- <MODULENAME>_AR_RELEASE_MINOR_VERSION
Where <MODULENAME> is the module abbreviation of the other (external) modules which provide header files included by the BswM module.

If the values are not identical to the expected values, an error shall be reported.

7.8 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 7.8.1. 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 needs 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 7.8.3 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 sent out 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 7.8.2 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 definitions. 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 on the usage:

```
SenderReceiverInterface AppModeRequestInterface {
  isService = true;
  AppModeRequestType requestedMode;
};
ModeSwitchInterface AppModeInterface {
  isService = true;
  AppModeType currentMode;
};
```

Figure 5 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.

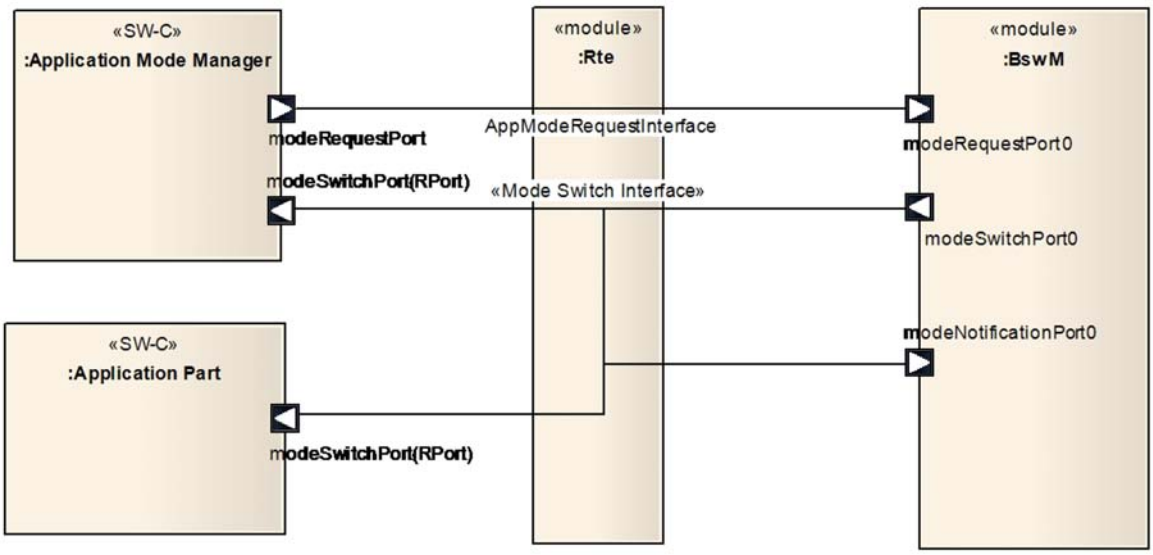


Figure 5: Connections between Application Mode Manager, Application Parts and the BSW Mode Manager

To actually switch the application mode, the BSW Mode Manager has a Mode Switch Port (modeSwitchPort0) 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. Here it is the Application Mode Manager, the other mode-dependent Application Part and the BSW Mode Manager itself (Note that it's named modeNotificationPort0 but the port type is Mode Switch Port). All of these connections are also local.

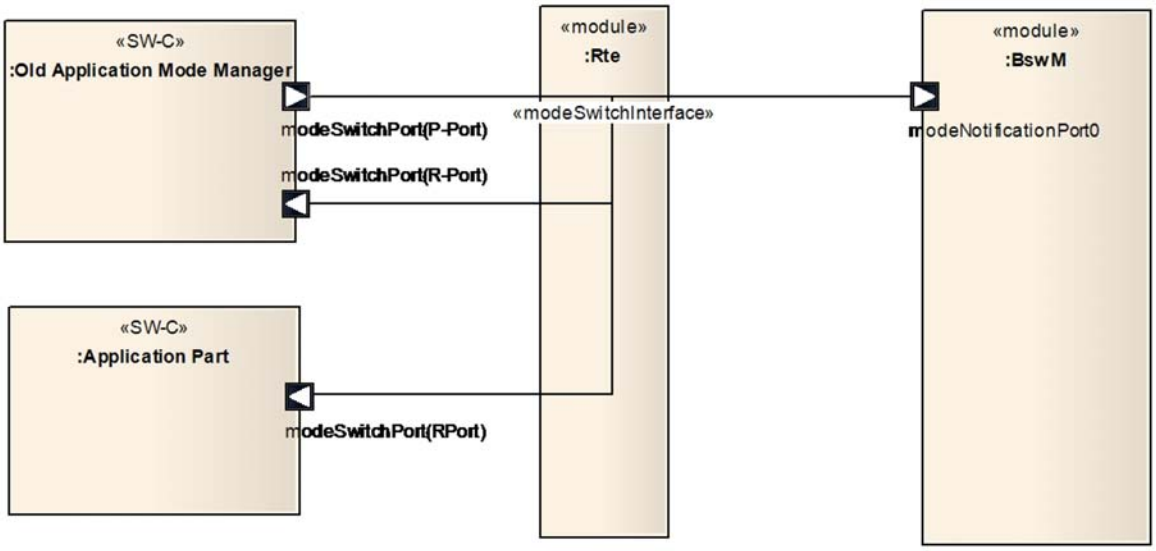


Figure 6: Connections between SW-C based Application Mode Manager, Application Parts and the BSW Mode Manager

Figure 6 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 made complete during ECU configuration time.

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

```
ARPackage AUTOSAR/Services/BswM
```

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

7.8.1 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 modeRequestPort0;
```

The configuration parameter `BswMSwcRequestPortRef` references this port.

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

```
Rte_Read_modeRequestPort0_requestedMode( &<variable> );
```

7.8.2 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 modeSwitchPort0;
```

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

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

```
Rte_Switch_modeSwitchPort0_currentMode( <new_mode> );

SchM_Switch_modeSwitchPort0_currentMode( <new_mode> );
```

7.8.3 Notifications of Mode Switches

In addition to mode requests, also the currently active modes can be 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 this document, to make it easier to distinguish between a RequirePort and ProvidePort of the ModeSwitchPort type, the RequirePorts are named mode notification port.

```
RequirePort AppModeInterface modeNotificationPort0;
```

The configuration parameter `BswMSwcModeNotificationInterfaceRef` references the Mode Switch interface used in this declaration.

To read the currently active mode, the BSW Mode Manager implementation has to call on of the following functions:

```
Rte_Mode_modeNotificationPort0_currentMode( &<variable> );

SchM_Mode_modeNotificationPort0_currentMode( &<variable> );
```

In case the enhanced Rte_Mode or SchM_Mode is configured, the BSW Mode Manager implementation has to call on of the following functions:

```
Rte_Mode_modeNotificationPort0_currentMode( &<variable>, &<previousmode>,
&<nextmode> );

SchM_Mode_modeNotificationPort0_currentMode( &<variable>, &<previosmode>,
&<nextmode> );
```

7.8.4 Component Type and Internal Behavior

The BSW Mode Manager is a Service Component that serves Mode Requests that are 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 has to do nothing special, as the BSWM Mode Manager reads the request cyclically in BswM_MainFunction. By contrast, for BSWM_IMMEDIATE the RTE has to 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:

BswM0138:
```
RunnableEntity ModeArbitrationRunnable {
  symbol = <mode_arbitration_function>;
  canBeInvokedConcurrently = TRUE;
};

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

Note: To deal with Mode Requests that originates from other ECU:s another kind of service component is needed. On 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 ServiceProxyComponentType 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.

7.8.5 BswM Service

This chapter summarizes the Software Component Description of the BSW Mode Manager.

[BswM0137] ⌜
```
ServiceComponentType BswM {

  // For each mode request
  RequirePort <request_interface> modeRequestPort<number>;

  // For each mode switch
  ProvidePort <mode_interface> modeSwitchPort<number>;

  // For each mode notification
  RequirePort <mode_interface> modeNotificationPort<number>;

  InternalBehavior {

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

```
    // For each mode request with BSWM_IMMEDIATE
    DataReceiveEvent ModeRequestEvent<number> {
      port = modeRequestPort<number>;
      dataElement = <data_element>;
      startOnEvent = ModeArbitrationRunnable;
    };

  };

};
```

()

8 API specification

8.1 Imported types

In this chapter, all types included from the following files are listed:

[BswM0001] ⌈The BSW Mode Manager shall use only the following imported types of other modules:

Module	*Imported Type*
CanSM	CanSM_BswMCurrentStateType
Com	Com_IpduGroupIdType
	Com_IpduGroupVector
ComM	ComM_InhibitionStatusType
	ComM_InitStatusType
	ComM_ModeType
	ComM_PncModeType
	ComM_UserHandleType
ComStack_Types	NetworkHandleType
	PNCHandleType
	PduIdType
Dcm	Dcm_CommunicationModeType
Dem	Dem_EventIdType
	Dem_EventStatusType
EcuM	EcuM_StateType
	EcuM_WakeupSourceType
	EcuM_WakeupStatusType
EthSm	EthSM_NetworkModeStateType
FrSm	FrSM_BswM_StateType
LinIf	LinIf_SchHandleType
	LinTp_Mode
LinSM	LinSM_ModeType
McOs	CoreIdType
NvM	NvM_BlockIdType
	NvM_RequestResultType
Os	ApplicationType
Std_Types	Std_ReturnType
	Std_VersionInfoType

⌋()

8.2 Type definitions

[BswM0041] ⌈The following Data Types shall be used for the functions defined in this specification.⌋()

8.2.1 BswM_ConfigType

Name:	`BswM_ConfigType`	
Type:	`Structure`	
Range:	`-`	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.	

[BswM0042] 「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」()

8.2.2 .BswM_ModeType

Name:	`BswM_ModeType`		
Type:	`uint8, uint16`		
Range:	`0-255, 0-65535`	`-` `-`	The range of valid IDs depends on configuration and on the chosen platform type.
Description:	This type identifies the modes that can be requested by BswM Users.		

8.2.3 BswM_UserType

Name:	`BswM_UserType`		
Type:	`uint8, uint16`		
Range:	`0-255, 0-65535`	`-` `-`	The range of valid IDs depends on configuration and on the chosen platform type.
Description:	This type identifies a BswM User that makes mode requests to the BswM.		

8.3 Function definitions

8.3.1 BswM_Init

[BswM0002] 「

Service name:	BswM_Init	

Syntax:	`void BswM_Init(`
	`  const BswM_ConfigType * ConfigPtr`
	`)`	

Service ID[hex]:	0x00	
Sync/Async:	Synchronous	
Reentrancy:	Non 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.

」(BSW101)

[BswM0043]「This routine initializes the BSW Mode Manager. After execution of this routine the BSW Mode Manager is ready to arbitrate incoming mode requests. 」()

[BswM0044]「This routine shall initialize all module global variables of the BSW Mode Manager.」()

[BswM0118]「BswM_Init shall only require the OS and the SchM to be initialized before it can be called.」()

[BswM0045]「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 Development Error Tracer with the value BSWM_E_PARAM_CONFIG.」()

[BswM0088]「
If the BswMDevErrorDetect switch is enabled and the configuration variant is VARIANT-POST-BUILD, the function BswM_Init shall check if a NULL pointer is passed for the ConfigPtr parameter. In case of an error the remaining function BswM_Init shall not be executed and the function BswM_Init shall report development error code BSWM_E_NULL_POINTER to the Det_ReportError service of the Development Error Tracer.」()

8.3.2 BswM_Deinit

[BswM0119]「

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.

」()

[BswM0120] ⌈
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. ⌋()

8.3.3 BswM_GetVersionInfo

[BswM0003] ⌈

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.	

⌋(BSW00407)

[BswM0004] ⌈
The function BswM_GetVersionInfo shall return the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers (BSW00407). ⌋()

[BswM0005] ⌈
The function BswM_GetVersionInfo shall be pre compile time configurable On/Off by the configuration parameter: BswMVersionInfoApi. ⌋()

[BswM0006] ⌈
If source code for caller and callee of BswM_GetVersionInfo is available, the BSW Mode Manager should realize BswM_GetVersionInfo as a macro, defined in the module's header file. ⌋()

[BswM0139] ⌈
If the BswMDevErrorDetect switch is enabled and the configuration variant is VARIANT-POST-BUILD, the function BswM_GetVersionInfo shall check if a NULL pointer is passed for the VersionInfo parameter. In case of an error the remaining function BswM_GetVersionInfo shall not be executed and the function BswM_GetVersionInfo shall report development error code

BSWM_E_PARAM_INVALID to the Det_ReportError service of the Development Error Tracer.」()

8.3.4 BswM_RequestMode

[BswM0046] 「

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.	

」(BSW09179, BSW09228)

[BswM0077] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0089] 「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0090] 「
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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

8.3.5 BswM_ComM_CurrentMode

[BswM0047] 「

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.	

」(BSW09228)

[BswM0078] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0091] 「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE. 」()

[BswM0092] 「
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

8.3.6 BswM_ComM_CurrentPNCMode

[BswM0148] 「

Service name:	BswM_ComM_CurrentPNCMode	

Syntax:	`void BswM_ComM_CurrentPNCMode(`
	`  PNCHandleType PNC,`
	`  ComM_PncModeType RequestedMode`
	`)`	

Service ID[hex]:	0x15	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	PNC	The handle of the PNC for which the current state is reported.
	RequestedMode	The current mode of the PNC.

Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Function called by ComM to indicate the current mode of the PNC.

⌋(BSW09228)

[BswM0149] ⌈
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 Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(BSW00406)

[BswM0150] ⌈
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE. ⌋()

[BswM0151] ⌈
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌋()

8.3.7 BswM_Dcm_CommunicationMode_CurrentState

[BswM0048] ⌈

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.	

⌋(BSW09228)

[BswM0079] ⌈

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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0093]「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0094]「
If the BswMDevErrorDetect switch is enabled, the parameter Network shall be checked for being in the allowed range. In case of error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

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_CanSM_CurrentState

[BswM0049]「

Service name:	BswM_CanSM_CurrentState	
Syntax:	`void BswM_CanSM_CurrentState( NetworkHandleType Network, CanSM_BswMCurrentStateType CurrentState )`	
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.	

」(BSW09228)

[BswM0080]「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0095]「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0096]「
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

8.3.9 BswM_EthSM_CurrentState

[BswM0050]「

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.	

」(BSW09228)

[BswM0081]「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0097]「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0098] ⌈
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌋()

8.3.10 BswM_FrSM_CurrentState

[BswM0051] ⌈

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.	

⌋(BSW09228)

[BswM0082] ⌈
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 Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(BSW00406)

[BswM0099] ⌈
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋()

[BswM0100] ⌈
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌋()

8.3.11 BswM_LinSM_CurrentState

[BswM0052] ⌜

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.	

⌟(BSW09228)

[BswM0083] ⌜
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 Development Error Tracer with the error code BSWM_E_NO_INIT.⌟(BSW00406)

[BswM0101] ⌜
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌟()

[BswM0102] ⌜
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌟()

8.3.12 BswM_EcuM_CurrentState

[BswM0056] ⌜

Service name:	BswM_EcuM_CurrentState
Syntax:	`void BswM_EcuM_CurrentState( EcuM_StateType CurrentState )`
Service ID[hex]:	0x0f
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 by EcuM to indicate the current ECU Operation Mode.	

」()

[BswM0084] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0103] 「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE and the routine shall return the value E_NOT_OK.」()

8.3.13 BswM_EcuM_CurrentWakeup

[BswM0131] 「

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.	

」()

[BswM0132] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0133] ⌈
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE and the routine shall return the value E_NOT_OK.⌋()

8.3.14 BswM_NvM_CurrentBlockMode

[BswM0104] ⌈

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. To use this function integration code will be needed.	

⌋()

[BswM0109] ⌈
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 Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(BSW00406)

[BswM0110] ⌈
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 block mode indication and report the error to the Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌋()

[BswM0111] ⌈
If the BswMDevErrorDetect switch is enabled, the parameter Block 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌋()

8.3.15 BswM_NvM_CurrentJobMode

[BswM0152] ⌜

Service name:	BswM_NvM_CurrentJobMode	

Syntax:	`void BswM_NvM_CurrentJobMode(`
	`  uint8 ServiceId,`
	`  NvM_RequestResultType CurrentJobMode`
	`)`	

Service ID[hex]:	0x17	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ServiceId	Indicates whether the callback refers to multi block services NvM_ReadAll or NvM_WriteAll.
	CurrentJobMode	Current state of the multi block job indicated by parameter ServiceId.
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.	

⌟()

[BswM0153] ⌜
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 Development Error Tracer with the error code BSWM_E_NO_INIT.⌟(BSW00406)

[BswM0154] ⌜
If the BswMDevErrorDetect switch is enabled, the parameter Block 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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.⌟()

[BswM0155] ⌜
If the BswMDevErrorDetect switch is enabled, the parameter Block 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.⌟()

8.3.16 BswM_LinSM_CurrentSchedule

[BswM0058] ⌜

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.	

」()

[BswM0086] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0107] 「
If the BswMDevErrorDetect switch is enabled, the parameter CurrentSchedule shall be checked for being in the allowed range. In case of an error the BswM shall ignore the schedule indication and report the error to the Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0108] 「
If the BswMDevErrorDetect switch is enabled, the parameter Network shall be checked for being in the allowed range. In case of an error, the BswM shall ignore the schedule indication and report the error to the Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

8.3.17 BswM_LinTp_RequestMode

[BswM0156] 「

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 mainly correlates to the LIN schedule table that should be used.

」()

[BswM0112] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0113] 「
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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

[BswM0114] 「
If the BswMDevErrorDetect switch is enabled, the parameter Network 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 Development Error Tracer with the value BSWM_E_REQ_USER_OUT_OF_RANGE.」()

8.3.18 BswM_WdgM_RequestPartitionReset

[BswM0157] 「

Service name:	BswM_WdgM_RequestPartitionReset
Syntax:	`void BswM_WdgM_RequestPartitionReset( ApplicationType Application )`
Service ID[hex]:	0x11
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	Application --
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Function called by WdgM to request a partition reset.

」()

[BswM0134] 「
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 Development Error Tracer with the error code BSWM_E_NO_INIT.」(BSW00406)

[BswM0135]「
If the BswMDevErrorDetect switch is enabled, the parameter Application 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 Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

8.3.19 BswM_TriggerStartUpPhase2

[BswM0140]「

Service name:	BswM_TriggerStartUpPhase2	
Syntax:	`Std_ReturnType BswM_TriggerStartUpPhase2( CoreIdType CoreID )`	
Service ID[hex]:	0x12	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	CoreID	The identifier of the slave core.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Task has been started correctly E_OS_CORE: Error during task creation
Description:	Function called by BswM to start phase two on a slave core.	

」()

[BswM0142]「
The function BswM_TriggerStartUpPhase2 starts a OS task on the slave core which starts the Scheduler Manager and the RTE.」()

[BswM0145]「
The function BswM_TriggerStartUpPhase2 shall return the result of the OS task creation.」()

8.3.20 BswM_TriggerSlaveRTEStop

[BswM0141]「

Service name:	BswM_TriggerSlaveRTEStop
Syntax:	`Std_ReturnType BswM_TriggerSlaveRTEStop( CoreIdType CoreID )`

Service ID[hex]:	0x13	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	CoreID	The identifier of the slave core.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Task has been started correctly E_OS_CORE: Error during task creation
Description:	Function called by BswM to stop the RTE of a slave core.	

」()

[BswM0143] 「
This function BswM_TriggerSlaveRTEStop starts a OS task on the slave core which stops the RTE.」()

[BswM0144] 「
The function BswM_TriggerSlaveRTEStop shall return the result of the OS task creation.」()

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

[BswM0053] 「

Service name:	BswM_MainFunction

Syntax:	`void BswM_MainFunction(`
	`  void`
	`)`

Service ID[hex]:	0x03
Timing:	FIXED_CYCLIC
Description:	Main function of the BswM

」()

[BswM0075] 「
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.」()

[BswM0076] ⌈
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. If the BswM-mainfunction is uninitialized called from the BSW Scheduler, then it shall return immediately without performing any action and without reporting an error.⌋()

8.6 Expected Interfaces

In this chapter, all 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.

[BswM0007] ⌈

API function	***Description***
Dem_ReportErrorStatus	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function.

⌋()

8.6.2 Optional Interfaces

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

[BswM0008] ⌈

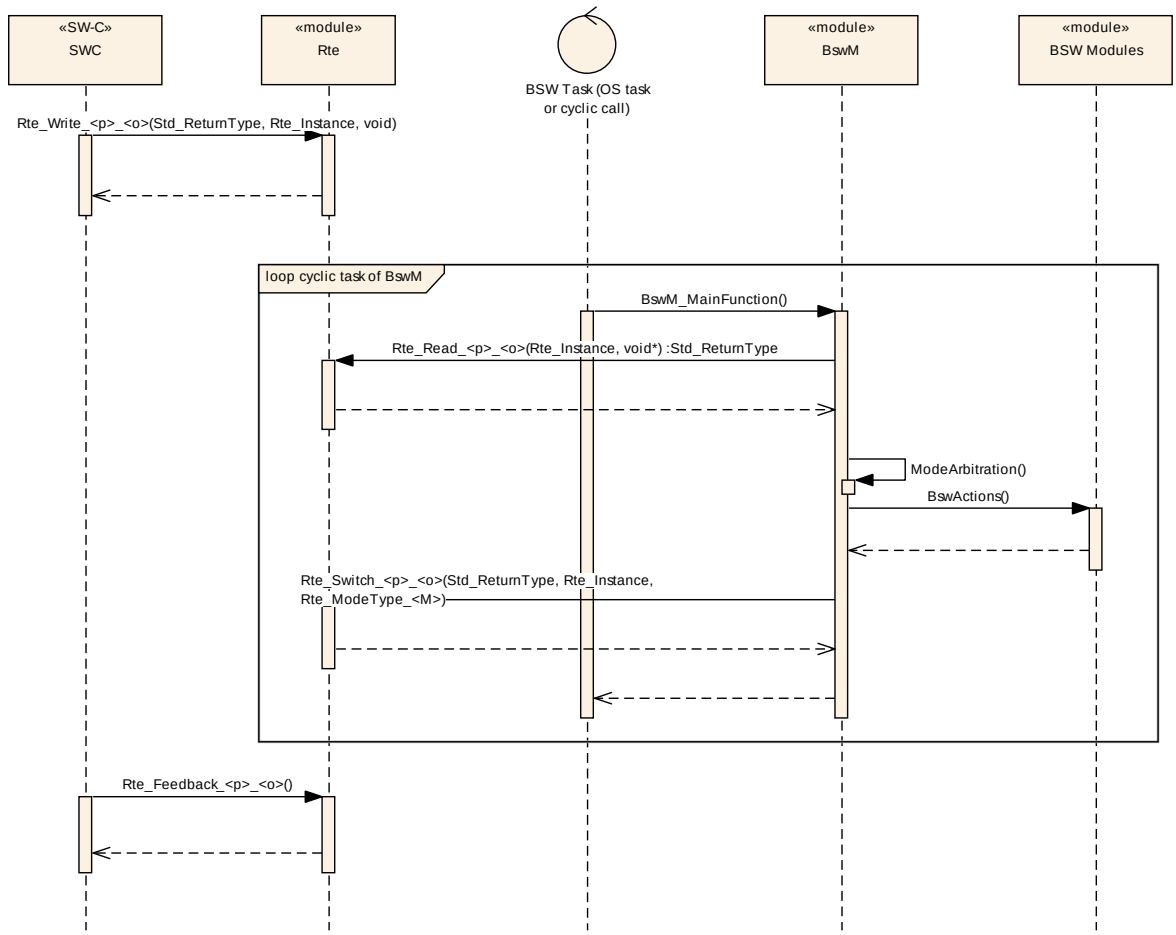
API function	***Description***
ComM_GetCurrentComMode	Function to query the current Communication Mode. ComM shall use the corresponding interfaces of the Bus State Managers to get the current Communication Mode of the network. (Call to Bus State Manager API: XXXSM _GetCurrentComMode(...))
ComM_GetInhibitionStatus	Returns the inhibition status of a ComM channel.
ComM_GetMaxComMode	Function to query the maximum allowed Communication Mode of the corresponding user.
ComM_GetRequestedComMode	Function to query the currently requested Communication Mode of the corresponding user.
ComM_GetStatus	Returns the initialization status of the AUTOSAR Communication Manager. After a call to ComM_DeInit() ComM should have status COMM_UNINIT, and a new call to ComM_Init needed to make sure ComM restart internal state machines to defailt values.
ComM_GetVersionInfo	This function returns the published information (for details refer to table 10.3)
ComM_LimitChannelToNoComMode	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	Changes the inhibition status for the ECU (=all channels) for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_LimitChannelToNoComMode, same functionality but for a specific channels)
ComM_MainFunction_<Channel_Id>	This function shall perform the processing of the AUTOSAR ComM activities that are not directly initiated by the calls e.g. from the RTE. There shall be one dedicated Main Function for each instance of ComM. Precondition: ComM shall be initialized
ComM_PreventWakeUp	Changes the inhibition status COMM_NO_WAKEUP for the corresponding channel.
ComM_ReadInhibitCounter	This function returns the amount of rejected COMM_FULL_COMMUNICATION user requests.
ComM_RequestComMode	Requesting of a Communication Mode by a user. Note: Internally mode COMM_SILENT_COMMUNICATION is not a valid request for a user, mode used for synchronization at shutdown. Valid modes are COMM_NO_COMMUNICATION and COMM_FULL_COMMUNICATION
ComM_ResetInhibitCounter	This function resets the Inhibited COMM_FULL_COMMUNICATION request Counter.
ComM_SetECUGroupClassification	Changes the ECU Group Classification status (see chapter 10.2.2)
Com_ClearIpduGroupVector	This service sets all bits of the given Com_IpduGroupVector to 0.
Com_IpduGroupControl	This service starts I-PDU groups.
Com_ReceptionDMControl	This service enables or disables I-PDU group Deadline Monitoring.
Com_SetIpduGroup	This service sets the value of a bit in an I-PDU group vector.
Com_SwitchIpduTxMode	The service Com_SwitchIpduTxMode sets the transmission mode of the I-PDU referenced by PduId to Mode. In case the transmission mode changes, the new mode shall immediately be effective (see COM239). In case the requested transmission mode was already active for this I-PDU, the call will have no effect.
Det_ReportError	Service to report development errors.
EcuM_GoDown	Instructs the ECU State Manager module to perform a power off or a reset depending on the selected shutdown target.
EcuM_SelectShutdownTarget	EcuM_SelectShutdownTarget selects the shutdown target. EcuM_SelectShutdownTarget is part of the ECU Manager Module port interface.
LinSM_ScheduleRequest	The upper layer requests a schedule table to be changed on one LIN network.
Nm_DisableCommunication	Disables the NM PDU transmission ability. For that purpose <BusNm>_DisableCommunication shall be called (e.g. CanNm_DisableCommunication function is called if channel is configured as CAN).
Nm_EnableCommunication	Enables the NM PDU transmission ability. For that purpose <BusNm>_EnableCommunication shall be called (e.g. CanNm_EnableCommunication function is called if channel is configured as CAN).

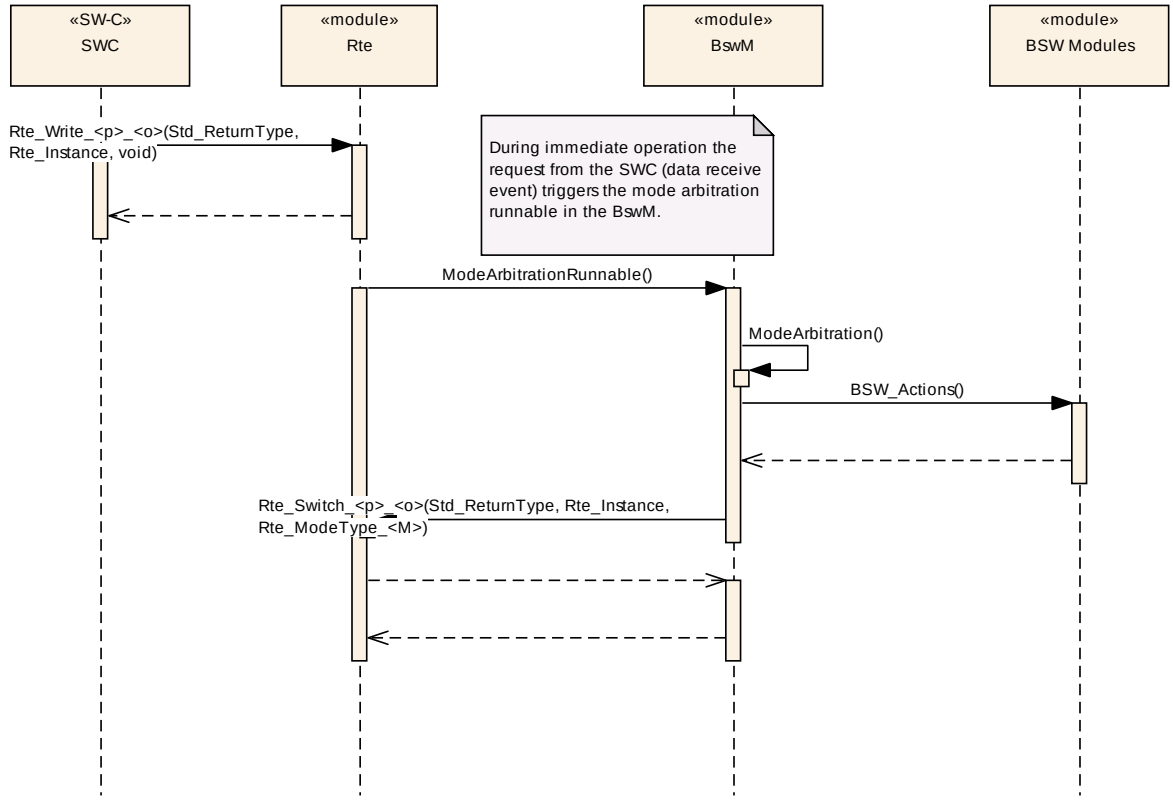
()

9 Sequence diagrams

9.1 Deferred operation of BswM



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

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture [2].
- AUTOSAR ECU Configuration Specification [8]. This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

10.1.2 Variants

The BSW Mode Manager provides three configuration variants.

[BswM0020] ⌈
Variant 1 – VARIANT-PRE-COMPILE: In this configuration variant all parameters need to be configured pre compile time.⌋(BSW00345)

[BswM0021] ⌜
Variant 2 - VARIANT-LINK-TIME: This configuration variant contains a mix of pre compile time and link time parameters. The configuration class of each parameter is defined in chapter 10.2.⌟(BSW00344)

10.1.3 Containers

Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

10.1.4 Specification template for configuration parameters

The following tables consist of three sections:

- the general section
- the configuration parameter section
- the section of included/referenced containers

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
<Reference a valid (sub)container by its name, e.g.,CanController>	*<Specifies the possible number of instances of the referenced container and its contained configuration parameters.* *Possible values:* *<multiplicity>* *<min_multiplicity..max_multiplicity>* *>*	*<Describe the scope of the referenced sub-container if known or mark it as "- -".* *The scope describes the impact of the configuration parameter: Does the setting affect only one instance of the module (instance), all instances of this module (module), the ECU or a network.* *Possible values of scope :* *instance, module, ECU, network>* *<Describe the dependencies with respect to the scope if known ot mark it as "- -".>*

Pre-compile time - specifies whether the configuration parameter shall be of configuration class *Pre-compile time* or not

Label	***Description***
X	The configuration parameter shall be of configuration class *Pre-compile time*.
--	The configuration parameter shall never be of configuration class *Pre-compile time*.

Link time - specifies whether the configuration parameter shall be of configuration class *Link time* or not

Label	***Description***
X	The configuration parameter shall be of configuration class *Link time*.

--	The configuration parameter shall never be of configuration class *Link time*.

Post Build - specifies whether the configuration parameter shall be of configuration class *Post Build* or not

Label	***Description***
X	The configuration parameter shall be of configuration class *Post Build* and no specific implementation is required.
L	*Loadable* - the configuration parameter shall be of configuration class *Post Build* and only one configuration parameter set resides in the ECU.
M	*Multiple* - the configuration parameter shall be of configuration class *Post Build* and is selected out of a set of multiple parameters by passing a dedicated pointer to the init function of the module.
--	The configuration parameter shall never be of configuration class *Post Build*.

10.2 Containers and configuration parameters

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

10.2.1 BswM

Module Name	*BswM*
Module Description	Configuration of the BswM (Basic SW Mode Manager) module.

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 is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
BswMGeneral	1	General configuration parameters of the Basic SW Mode Manager.

10.2.2 BswMGeneral

SWS Item	**BswM0800_Conf :**
Container Name	BswMGeneral
Description	General configuration parameters of the Basic SW Mode Manager.
Configuration Parameters	

SWS Item	**BswM00938_Conf :**		
Name	BswMCanSMEnabled {BSWM_CANSM_ENABLED}		
Description	enable/disable CanSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**BswM00939_Conf :**		
Name	BswMComMEnabled {BSWM_COMM_ENABLED}		
Description	enable/disable ComM module related BswM API: true: Enabled false: Disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClas	***Pre-compile time***	X	All Variants

s			
	Link time	--	
	Post-build time	--	
Scope Dependency	/scope: Module		

SWS Item	**BswM00940_Conf :**		
Name	BswMDcmEnabled {BSWM_DCM_ENABLED}		
Description	enable/disable Dcm module related BswM API: true: Enabled false: Disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope Dependency	/scope: Module		

SWS Item	**BswM0811_Conf :**		
Name	BswMDevErrorDetect {BSWM_DEV_ERROR_DETECT}		
Description	Switches the Development Error Detection and Notification ON or OFF. true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope Dependency	/scope: Module		

SWS Item	**BswM00941_Conf :**		
Name	BswMEcuMEnabled {BSWM_ECUM_ENABLED}		
Description	enable/disable EcuM module related BswM API: true: Enabled false: Disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope Dependency	/scope: Module		

SWS Item	**BswM00942_Conf :**		
Name	BswMEthSMEnabled {BSWM_ETHSM_ENABLED}		
Description	enable/disable EthSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	

Scope Dependency	/scope: Module

SWS Item	**BswM00943_Conf :**			
Name	BswMFrSMEnabled {BSWM_FRSM_ENABLED}			
Description	enable/disable FrSM module related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope Dependency	/scope: Module			

SWS Item	**BswM00949_Conf :**			
Name	BswMGenericRequestEnabled {BSWM_GENERIC_REQUEST_ENABLED}			
Description	enable/disable Generic Request related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope Dependency	/scope: Module			

SWS Item	**BswM00944_Conf :**			
Name	BswMLinSMEnabled {BSWM_LINSM_ENABLED}			
Description	enable/disable LinSM module related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope Dependency	/scope: Module			

SWS Item	**BswM00945_Conf :**			
Name	BswMLinTPEnabled {BSWM_LINTP_ENABLED}			
Description	enable/disable LinTP module related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope Dependency	/scope: Module			

SWS Item	**BswM0813_Conf :**

Name	BswMMainFunctionPeriod			
Description	The cycle time of the periodic main function of BswM. Defined in seconds .			
Multiplicity	0..1			
Type	EcucFloatParamDef			
Range	`0 .. INF`			
Default value	--			
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE	
	Link time	X	VARIANT-LINK-TIME	
	Post-build time	--		
Scope / Dependency	scope: Module			

SWS Item	**BswM00946_Conf :**			
Name	BswMNvMEnabled {BSWM_NVM_ENABLED}			
Description	enable/disable NvM module related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope / Dependency	scope: Module			

SWS Item	**BswM00947_Conf :**			
Name	BswMSchMEnabled {BSWM_SCHM_ENABLED}			
Description	enable/disable SchM module related BswM API: true: Enabled false: Disabled			
Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope / Dependency	scope: Module			

SWS Item	**BswM0812_Conf :**			
Name	BswMVersionInfoApi			
Description	Switches the possibility to read the version information with the service BswM_GetVersionInfo(). true: Enabled false: Disabled			
Multiplicity	1			
Type	EcucBooleanParamDef			
Default value	true			
ConfigurationClass	***Pre-compile time***	X	All Variants	
	Link time	--		
	Post-build time	--		
Scope / Dependency	scope: Module			

SWS Item	**BswM00948_Conf :**
Name	BswMWdgMEnabled {BSWM_WDGM_ENABLED}
Description	enable/disable WdgM module related BswM API: true: Enabled false: Disabled

Multiplicity	0..1			
Type	EcucBooleanParamDef			
Default value	false			
ConfigurationClass	***Pre-compile time***		X	All Variants
	Link time	--		
	Post-build time	--		
Scope / Dependency	scope: Module			

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

10.2.3 BswMUserIncludeFiles

SWS Item	**BswM00954_Conf :**
Container Name	BswMUserIncludeFiles
Description	Collection of header file names which shall be included by the BswM.
Configuration Parameters	

SWS Item	**BswM00955_Conf :**		
Name	BswMUserIncludeFile		
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 C90 section 6.10.2 "source file inclusion". The parameter value MUST NOT represent a path, since ISO C90 does not specify how such a path is treated (i.e., this is implementation defined (and additionally depends on the operating system and the underlying file system)).		
Multiplicity	1..*		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.4 BswMConfig

SWS Item	**BswM0895_Conf :**
Container Name	BswMConfig
Description	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
Configuration Parameters	

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.

10.2.5 BswMArbitration

SWS Item	**BswM0801_Conf :**
Container Name	BswMArbitration
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***
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. 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.

10.2.6 BswMRule

SWS Item	**BswM0806_Conf :**
Container Name	BswMRule
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.
Configuration Parameters	

SWS Item	**BswM00935_Conf :**		
Name	BswMNestedExecutionOnly		
Description	This parameter defines for its related Rule if the Rule is an Independent rule or a Subordinate rule; 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). 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		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**BswM0888_Conf :**		
Name	BswMRuleInitState		
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. 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. If this parameter is set to BSWM_TRUE, the evaluation result is treated as changed if the rule is evaluated to false. 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	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	scope: Instance

SWS Item	**BswM0819_Conf :**		
Name	BswMRuleExpressionRef		
Description	This choice reference either references the mode condition or the logical expression that is evaluated for each rule.		
Multiplicity	1		
Type	Reference to [BswMLogicalExpression]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0818_Conf :**		
Name	BswMRuleFalseActionList		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0817_Conf :**		
Name	BswMRuleTrueActionList		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.7 BswMModeRequestPort

SWS Item	**BswM0805_Conf :**
Container Name	BswMModeRequestPort
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. Note that the BswM treats all request and indications in the exact same way.
Configuration Parameters	

SWS Item	**BswM0822_Conf :**		
Name	BswMRequestProcessing		
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	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Module		

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. This container is optional and shall only be used for Mode Requests that do not already have an initial value. If this container is not present the requested mode is undefined after initialization of BswM. The requested mode will remain undefined until the requester performs a request.
BswMModeRequestSource	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.

10.2.8 BswMModeInitValue

SWS Item	**BswM0928_Conf :**
Choice container Name	BswMModeInitValue
Description	This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. This container is optional and shall only be used for Mode Requests that do not already have an initial value. If this container is not present the requested mode is undefined after initialization of BswM. The requested mode will remain undefined until the requester performs a request.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMBswModeInitValue	0..1	This is the choice for a initial mode value used for the initialization of mode requests in case the request is made by a BSW module.
BswMSwcModeInitValue	0..1	This is the choice for a foreign reference to the Mode Declaration used for the initialization of mode requests in case the request is made by a SW-C.

10.2.9 BswMBswModeInitValue

SWS Item	**BswM0931_Conf :**
Container Name	BswMBswModeInitValue
Description	This is the choice for a initial mode value used for the initialization of mode requests in case the request is made by a BSW module.
Configuration Parameters	

SWS Item	**BswM0932_Conf :**		
Name	BswMBswModeInitValueMode		
Description	This parameter defines the initial mode value that is used by BswM for the corresponding mode request after initialization.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.10 BswMSwcModeInitValue

SWS Item	**BswM0929_Conf :**
Container Name	BswMSwcModeInitValue
Description	This is the choice for a foreign reference to the Mode Declaration used for the initialization of mode requests in case the request is made by a SW-C.
Configuration Parameters	

SWS Item	**BswM0930_Conf :**		
Name	BswMSwcModeInitValueRef		
Description	This is a foreign reference to the Mode Declaration used for the initialization of mode requests.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.11 BswMModeRequestSource

SWS Item	**BswM0856_Conf :**
Choice container Name	BswMModeRequestSource
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.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMBswModeNotification	0..1	The source of the mode request is a Bsw Module. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_IMMEDIATE then it is assumed that the Basic Software Module Description of the BswM contains a BswSchedulableEntity which is activated by a BswModeSwitchEvent. This BswModeSwitchEvent shall refer to the ModeDeclarationGroupPrototype which is referenced by BswMBswModeDeclarationGroupPrototypeRef. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_DEFERRED then it is up to the implementer of the BswM tooling whether a BswSchedulableEntity is used to update the BswM internal mode mirror or whether the BswM internal mode mirror is updated during the main function execution.
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.
BswMDcmApplicationUpdatedIndication	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.
BswMDcmComModeRequest	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.
BswMEcuMWakeupSource	0..1	This is a notification of the current state of an ECU State Manager wakeup source.
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).
BswMLinSMIndication	0..1	This is an indication of the current state of the LIN State Manager.
BswMLinScheduleIndication	0..1	This is an indication of the currently active LIN

		Schedule Table for a specific LIN Interface.
BswMLinTpModeRequest	0..1	This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API.
BswMNvMJobModeIndication	0..1	Indicates the current status of the multiblock job. The job is identified via BswMNvmService, e.g. 0x0c for NvmReadAll, 0x0d for NvmWriteAll. 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
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
BswMSwcModeNotification	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.
BswMWdgMRequestPartitionReset	0..1	This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.

10.2.12 BswMBswModeNotification

SWS Item	**BswM0926_Conf :**
Container Name	BswMBswModeNotification
Description	The source of the mode request is a Bsw Module. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_IMMEDIATE then it is assumed that the Basic Software Module Description of the BswM contains a BswSchedulableEntity which is activated by a BswModeSwitchEvent. This BswModeSwitchEvent shall refer to the ModeDeclarationGroupPrototype which is referenced by BswMBswModeDeclarationGroupPrototypeRef. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_DEFERRED then it is up to the implementer of the BswM tooling whether a BswSchedulableEntity is used to update the BswM internal mode mirror or whether the BswM internal mode mirror is updated during the main function execution.
Configuration Parameters	

SWS Item	**BswM0927_Conf :**
Name	BswMBswModeDeclarationGroupPrototypeRef

Description	This is a foreign reference to the Mode Declaration Group Prototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.13 BswMCanSMIndication

SWS Item	**BswM0857_Conf :**
Container Name	BswMCanSMIndication
Description	This is an indication of the current state of the CAN State Manager.
Configuration Parameters	

SWS Item	**BswM0870_Conf :**		
Name	BswMCanSMChannelRef		
Description	This is a reference to the CAN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.14 BswMDcmComModeRequest

SWS Item	**BswM0863_Conf :**
Container Name	BswMDcmComModeRequest
Description	The source of the mode request is the Diagnostic Communication Manager.
Configuration Parameters	

SWS Item	**BswM0876_Conf :**		
Name	BswMDcmComMNetwork		
Description	This parameter specifies the network the request relates to.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME

	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.15 BswMEcuMIndication

SWS Item	**BswM0879_Conf :**
Container Name	BswMEcuMIndication
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

10.2.16 BswMEcuMWakeupSource

SWS Item	**BswM0904_Conf :**
Container Name	BswMEcuMWakeupSource
Description	This is a notification of the current state of an ECU State Manager wakeup source.
Configuration Parameters	

SWS Item	**BswM0905_Conf :**		
Name	BswMEcuMWakeupSrcRef		
Description	This is a reference to the ECU State Manager Wakeup Source that the indication corresponds to.		
Multiplicity	1		
Type	Reference to [EcuMWakeupSource]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.17 BswMEthSMIndication

SWS Item	**BswM0860_Conf :**
Container Name	BswMEthSMIndication
Description	This is an indication of the current state of the Ethernet State Manager.
Configuration Parameters	

SWS Item	**BswM0873_Conf :**		
Name	BswMEthSMChannelRef		
Description	This is a reference to the Ethernet channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.18 BswMFrSMIndication

SWS Item	**BswM0858_Conf :**
Container Name	BswMFrSMIndication
Description	This is an indication of the current state of the FlexRay State Manager.
Configuration Parameters	

SWS Item	**BswM0872_Conf :**		
Name	BswMFrSMChannelRef		
Description	This is a reference to the FlexRay Cluster handle that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.19 BswMComMIndication

SWS Item	**BswM0880_Conf :**
Container Name	BswMComMIndication
Description	This is an indication of the current communication mode of a channel in the Communication Manager.
Configuration Parameters	

SWS Item	**BswM0883_Conf :**
Name	BswMComMChannelRef
Description	This is a reference to the Communication Manager channel handle that the indication corresponds to.
Multiplicity	1
Type	Reference to [ComMChannel]

ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.20 BswMComMPncRequest

SWS Item	**BswM0922_Conf :**
Container Name	BswMComMPncRequest
Description	This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager.
Configuration Parameters	

SWS Item	**BswM0924_Conf :**		
Name	BswMComMPncRef		
Description	This is a reference to the Communication Manager PNC handle of the Partial Network Cluster that the request corresponds to.		
Multiplicity	1		
Type	Reference to [ComMPnc]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.21 BswMNvMJobModeIndication

SWS Item	**BswM0956_Conf :**
Container Name	BswMNvMJobModeIndication
Description	Indicates the current status of the multiblock job. The job is identified via BswMNvmService, e.g. 0x0c for NvmReadAll, 0x0d for NvmWriteAll. 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	

SWS Item	**BswM0957_Conf :**	
Name	BswMNvmService	
Description	Identifies the Nvm job which is related to the mode request.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	NvmReadAll	NvmReadAll corresponds to service Id 0x0c
	NvmWriteAll	NvmWriteAll

		corresponds to service Id 0x0d	
ConfigurationClass	***Pre-compile time***	--	
	Link time	--	
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.22 BswMNvMRequest

SWS Item	**BswM0890_Conf :**
Container Name	BswMNvMRequest
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	

SWS Item	**BswM0891_Conf :**		
Name	BswMNvMBlockRef		
Description	This is a reference to the NvM Block Descriptor that the request corresponds to.		
Multiplicity	1		
Type	Reference to [NvMBlockDescriptor]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

BswMLinScheduleIndication

SWS Item	**BswM0885_Conf :**
Container Name	BswMLinScheduleIndication
Description	This is an indication of the currently active LIN Schedule Table for a specific LIN Interface.
Configuration Parameters	

SWS Item	**BswM0886_Conf :**		
Name	BswMLinScheduleRef		
Description	This is a reference to the LIN Schedule Table handle that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [LinSMSchedule]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.23 BswMLinTpModeRequest

SWS Item	**BswM0914_Conf :**
Container Name	BswMLinTpModeRequest
Description	This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API.
Configuration Parameters	

SWS Item	**BswM0915_Conf :**		
Name	BswMLinTpChannelRef		
Description	This is a reference to the LIN Interface Channel that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [LinIfChannel]		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.24 BswMGenericRequest

SWS Item	**BswM0861_Conf :**
Container Name	BswMGenericRequest
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	

SWS Item	**BswM0874_Conf :**		
Name	BswMModeRequesterId		
Description	This parameters identifies the different users of the generic mode request interface.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0875_Conf :**
Name	BswMRequestedModeMax
Description	This parameter defines the upper limit for the modes requested by this mode requester.

Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.25 BswMLinSMIndication

SWS Item	**BswM0859_Conf :**
Container Name	BswMLinSMIndication
Description	This is an indication of the current state of the LIN State Manager.
Configuration Parameters	

SWS Item	**BswM0871_Conf :**		
Name	BswMLinSMChannelRef		
Description	This is a reference to the LIN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.26 BswMSwcModeRequest

SWS Item	**BswM0862_Conf :**
Container Name	BswMSwcModeRequest
Description	The source of the mode request is a SW Component.
Configuration Parameters	

SWS Item	**BswM0934_Conf :**		
Name	BswMSwcModeRequestModeDeclarationGroupPrototypeRef		
Description	This is a foreign reference to the ModeDeclarationGroupPrototype. As the mode request is SR-Communication the BswM shall provide a SR-Interface which corresponds to the ModeDeclarationGroupPrototype. The SR-Interface shall contain one ApplicationPrimitiveDataType which is defined as an enumeration (compuMethod) with the enumeration literals matching the ModeDeclarations.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE

	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.27 BswMSwcModeNotification

SWS Item	**BswM0892_Conf :**
Container Name	BswMSwcModeNotification
Description	This is a mode switch notification associated with a RTE switch interface.
Configuration Parameters	

SWS Item	**BswM0893_Conf :**		
Name	BswMSwcModeNotificationModeDeclarationGroupPrototypeRef		
Description	This is a foreign reference to the ModeDeclarationGroupPrototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.28 BswMWdgMRequestPartitionReset

SWS Item	**BswM0916_Conf :**
Container Name	BswMWdgMRequestPartitionReset
Description	This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.
Configuration Parameters	

SWS Item	**BswM0917_Conf :**		
Name	BswMWdgMRequestPartitionResetRef		
Description	This is a reference to the partition that shall be reset.		
Multiplicity	1		
Type	Reference to [EcucPartition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.29 BswMModeCondition

SWS Item	**BswM0807_Conf :**
Container Name	BswMModeCondition
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.
Configuration Parameters	

SWS Item	**BswM0815_Conf :**		
Name	BswMConditionType		
Description	This parameter specifies what kind of comparison that is made for the evaluation of the mode condition.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_EQUALS	--	
	BSWM_EQUALS_NOT	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0821_Conf :**		
Name	BswMConditionMode		
Description	This parameter references the mode request port that is used for the condition.		
Multiplicity	1		
Type	Reference to [BswMModeRequestPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

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.

10.2.30 BswMConditionValue

SWS Item	**BswM0816_Conf :**
Choice container Name	BswMConditionValue
Description	This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMBswMode	0..1	This container defines the value and type of a mode in the BSW.
BswMModeDeclaratio	0..1	When the mode corresponds to a mode request or mode indication

n		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.

10.2.31 BswMBswMode

SWS Item	**BswM0869_Conf :**
Container Name	BswMBswMode
Description	This container defines the value and type of a mode in the BSW.
Configuration Parameters	

SWS Item	**BswM0866_Conf :**		
Name	BswMBswRequestedMode		
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	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.32 BswMModeDeclaration

SWS Item	**BswM0868_Conf :**
Container Name	BswMModeDeclaration
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	

SWS Item	**BswM0864_Conf :**		
Name	BswMModeValueRef		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	

No Included Containers

10.2.33 BswMLogicalExpression

SWS Item	**BswM0808_Conf :**
Container Name	BswMLogicalExpression
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.
Configuration Parameters	

SWS Item	**BswM0814_Conf :**		
Name	BswMLogicalOperator		
Description	This parameter specifies the logical operator to be used in the logical expression. If the expression only consists of a single condition this parameter shall not be used.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BSWM_AND	--	
	BSWM_NAND	--	
	BSWM_OR	--	
	BSWM_XOR	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0820_Conf :**		
Name	BswMArgumentRef		
Description	This is a choice reference either to a mode condition or a sub-expression. In case the BswMLogicalExpression.BswMLogicalOperator equals BSWM_NAND only two operands are supported.		
Multiplicity	1..*		
Type	Choice reference to [BswMLogicalExpression , BswMModeCondition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.34 BswMDataTypeMappingSets

SWS Item	**BswM00936_Conf :**

Container Name	BswMDataTypeMappingSets
Description	Collection of references to DataTypeMappingSet.
Configuration Parameters	

SWS Item	**BswM0937_Conf :**		
Name	BswMDataTypeMappingSetRef		
Description	Reference to DataTypeMappingSet.		
Multiplicity	1..*		
Type	Foreign reference to [DATA-TYPE-MAPPING-SET]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.35 BswMModeControl

SWS Item	**BswM0802_Conf :**
Container Name	BswMModeControl
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.
BswMSwitchPort	0..*	This container specifies PPorts and/or providedModeDeclarationGroups, which the BswM has to create in its SWCD resp. BSWMD. If the container is referenced by one or more BswMRteSwitchActions the BswM shall create a corresponding PPort in its Service Description. If the container is referenced by a BswMSchMSwitch action the BswM shall create the corresponding ModeDeaclarationGroupPrototype as providedModeDeclarationGroup in it BSWMD. If the container is referenced by BswMSchMSwitch AND BswmRteSwitchActions the a providedModeDeclarationGroup as well as a PPort shall be created. In the corresponding SwcBswMapping a synchronizedModeGroup has to be created. See also chapter 4.4.7 in SWS_RTE.

10.2.36 BswMAction

SWS Item	**BswM0810_Conf :**
Container Name	BswMAction
Description	Each container of this type defines an action. These actions

	can be part of one or several action lists.
Configuration Parameters	

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

10.2.37 BswMActionList

SWS Item	**BswM0809_Conf :**
Container Name	BswMActionList
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.
Configuration Parameters	

SWS Item	**BswM0894_Conf :**		
Name	BswMActionListExecution		
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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_CONDITION	--	
	BSWM_TRIGGER	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

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

10.2.38 BswMActionListItem

SWS Item	**BswM0823_Conf :**
Container Name	BswMActionListItem
Description	This container defines an item in an action list.
Configuration Parameters	

SWS Item	**BswM0902_Conf :**
Name	BswMAbortOnFail
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		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0824_Conf :**		
Name	BswMActionListItemIndex		
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	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0825_Conf :**		
Name	BswMActionListItemRef		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0933_Conf :**		
Name	BswMReportFailToDemRef		
Description	If the reference is given, the DEM event shall be reported failed if this specific action returns E_NOT_OK; it shall be reported passed if this specific action returns E_OK.		
Multiplicity	0..1		
Type	Reference to [DemEventParameter]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.39 BswMAvailableActions

SWS Item	**BswM0826_Conf :**

Choice container Name	BswMAvailableActions
Description	Choice container including the available actions to be used in the action lists.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
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.
BswMComMModeLimitation	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.
BswMDeadlineMonitoringControl	0..1	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM. COM_ReceptionDMControl is called when this action is configured.
BswMEcuMGoDown	0..1	This container defines the UserId which shall be forwarded to the GoDown request.
BswMEcuMSelectShutdownTarget	0..1	This container defines the shutdown target.
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. 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. Com_IpduGroupControl is called when this action is configured.
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.
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.
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.
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.
BswMTriggerSlaveRTEStop	0..1	This container includes all parameters needed to stop the RTE on a slave core. This choice shall only be chosen if multicore is used.
BswMTriggerStartUpPhase2	0..1	This container includes all parameters needed to start phase two on a slave core. This choice shall only be chosen if multicore is used.
BswMUserCallout	0..1	This container includes all details needed for a user defined function call.

10.2.40 BswMSwitchPort

SWS Item	**BswM0950_Conf :**
Container Name	BswMSwitchPort
Description	This container specifies PPorts and/or providedModeDeclarationGroups, which the BswM has to create in its SWCD resp. BSWMD. If the container is referenced by one or more BswMRteSwitchActions the BswM shall create a corresponding PPort in its Service Description. If the container is referenced by a BswMSchMSwitch action the BswM shall create the corresponding ModeDeaclarationGroupPrototype as providedModeDeclarationGroup in it BSWMD. If the container is referenced by BswMSchMSwitch AND BswmRteSwitchActions the a providedModeDeclarationGroup as well as a PPort shall be created. In the corresponding SwcBswMapping a synchronizedModeGroup has to be created. See also chapter 4.4.7 in SWS_RTE.
Configuration Parameters	

SWS Item	**BswM0951_Conf :**		
Name	BswMModeSwitchInterfaceRef		
Description	Reference to the ModeSwitchInterface of this BswMModeSwitchPort.		
Multiplicity	1		
Type	Foreign reference to [MODE-SWITCH-INTERFACE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.41 BswMComMModeSwitch

SWS Item	**BswM0831_Conf :**
Container Name	BswMComMModeSwitch
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	

SWS Item	**BswM0840_Conf :**		
Name	BswMComMRequestedMode		
Description	This parameter specifies if the requested communication mode. This parameter corresponds to the parameter "ComMode" of the function ComM_RequestComMode.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_FULL_COM	--	
	BSWM_NO_COM	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0841_Conf :**		
Name	BswMComMUserRef		
Description	This is a reference to the ComM User that is associated to the Communication channel for which the communication mode should be requested. This reference corresponds to the parameter "User" of the function ComM_RequestComMode.		
Multiplicity	1		
Type	Reference to [ComMUser]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.42 BswMComMAllowCom

SWS Item	**BswM0909_Conf :**
Container Name	BswMComMAllowCom
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	

SWS Item	**BswM0918_Conf :**
Name	BswMComAllowed
Description	The parameter BswMComMAllowChannelRef refers to a channel which will allow or block communication using the function ComM_CommunicationAllowed(). This parameter corresponds to the parameter "Allowed" of the function ComM_CommunicationAllowed().
Multiplicity	1
Type	EcucBooleanParamDef

Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0912_Conf :**		
Name	BswMComMAllowChannelRef		
Description	This is a reference to the ComM Channel for which communication shall be allowed or blocked. This reference corresponds to the parameter "Channel" of the function ComM_CommunicationAllowed().		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.43 BswMComMModeLimitation

SWS Item	**BswM0908_Conf :**
Container Name	BswMComMModeLimitation
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	

SWS Item	**BswM0910_Conf :**		
Name	BswMComMLimitMode		
Description	The function ComM_LimitChannelToNoComMode() takes in this boolean parameter to limit the channel's com mode to no-com mode. This parameter corresponds to the parameter "Status" of the function ComM_LimitChannelToNoComMode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0911_Conf :**		
Name	BswMComMLimitChannelRef		
Description	This is a reference to the ComM channel for which the communication mode should be limited. This reference corresponds to the parameter "Channel" of the function ComM_LimitChannelToNoComMode.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE

	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.44 BswMDeadlineMonitoringControl

SWS Item	**BswM0830_Conf :**
Container Name	BswMDeadlineMonitoringControl
Description	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM. COM_ReceptionDMControl is called when this action is configured.
Configuration Parameters	

SWS Item	**BswM0852_Conf :**		
Name	BswMDisabledDMPduGroupRef		
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be disabled. Together with the BswMEnabledDMPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function COM_ReceptionDMControl.		
Multiplicity	0..*		
Type	Reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0851_Conf :**		
Name	BswMEnabledDMPduGroupRef		
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be enabled. Together with the BswMDisabledDMPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function COM_ReceptionDMControl.		
Multiplicity	0..*		
Type	Reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.45 BswMTriggerIPduSend

SWS Item	**BswM0906_Conf :**
Container Name	BswMTriggerIPduSend

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	

SWS Item	**BswM0907_Conf :**		
Name	BswMTriggeredIPduRef		
Description	This is a reference to an I-PDU that should be triggered for transmission. This reference corresponds to the parameter "PduId" of the function Com_TriggerIPDUSend.		
Multiplicity	0..*		
Type	Reference to [ComIPdu]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.46 BswMEcuMGoDown

SWS Item	**BswM0963_Conf :**
Container Name	BswMEcuMGoDown
Description	This container defines the UserId which shall be forwarded to the GoDown request.
Configuration Parameters	

SWS Item	**BswM0964_Conf :**		
Name	BswMEcuMUserIdRef		
Description	This is a reference to a EcuM UserId.		
Multiplicity	1		
Type	Reference to [EcuMFlexUserConfig]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.47 BswMEcuMSelectShutdownTarget

SWS Item	**BswM0961_Conf :**
Container Name	BswMEcuMSelectShutdownTarget
Description	This container defines the shutdown target.
Configuration Parameters	

SWS Item	**BswM0962_Conf :**
Name	BswMEcuMShutdownTargetRef
Description	This is a reference to a shutdown target.

Multiplicity	1		
Type	Reference to [EcuMSleepMode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.48 BswMNMControl

SWS Item	**BswM0837_Conf :**
Container Name	BswMNMControl
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	

SWS Item	**BswM0838_Conf :**		
Name	BswMNMAction		
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	--	
	BSWM_NM_ENABLE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0xxx_Conf :**		
Name	BswMComMNetworkHandleRef		
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId. This reference corresponds to the parameter "NetworkHandle" of the function Nm_EnableCommunication and Nm_DisableCommunication.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.49 BswMTriggerIPduSend

SWS Item	**BswM0906_Conf :**
Container Name	BswMTriggerIPduSend
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	

SWS Item	**BswM0907_Conf :**		
Name	BswMTriggeredIPduRef		
Description	This is a reference to an I-PDU that should be triggered for transmission. This reference corresponds to the parameter "PduId" of the function Com_TriggerIPDUSend.		
Multiplicity	0..*		
Type	Reference to [ComIPdu]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.50 BswMNMControl

SWS Item	**BswM0837_Conf :**
Container Name	BswMNMControl
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	

SWS Item	**BswM0838_Conf :**		
Name	BswMNMAction		
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	--	
	BSWM_NM_ENABLE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0xxx_Conf :**
Name	BswMComMNetworkHandleRef
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel

	index value in ComMChannelId. This reference corresponds to the parameter "NetworkHandle" of the function Nm_EnableCommunication and Nm_DisableCommunication.		
Multiplicity	1		
Type	Reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.51 BswMLinScheduleSwitch

SWS Item	**BswM0827_Conf :**
Container Name	BswMLinScheduleSwitch
Description	This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured. 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	

SWS Item	**BswM0842_Conf :**		
Name	BswMLinScheduleRef		
Description	This is a reference to the LIN schedule table that the LIN SM shall change to. This reference corresponds to the parameter "schedule" of the function LinSM_ScheduleRequest.		
Multiplicity	1		
Type	Reference to [LinSMSchedule]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.52 BswMPduGroupSwitch

SWS Item	**BswM0828_Conf :**
Container Name	BswMPduGroupSwitch
Description	This container includes references to the PDU groups that shall be enabled and disabled. Com_IpduGroupControl is called when this action is configured.
Configuration Parameters	

SWS Item	**BswM0913_Conf :**
Name	BswMPduGroupSwitchReinit

Description	This parameter defines if the values for the parameters like periodical timer, minimum delay timer etc is retainer or reinitialized during a PDU Group Switch. This parameter corresponds to the parameter "initialize" of the function Com_IpduGroupControl.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0850_Conf :**		
Name	BswMDisabledPduGroupRef		
Description	This is a reference to a PDU Group that should be disabled. Together with the BswMEnabledIPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function Com_IpduGroupControl.		
Multiplicity	0..*		
Type	Reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0849_Conf :**		
Name	BswMEnabledPduGroupRef		
Description	This is a reference to a PDU Group that should be enabled. Together with the BswMDisabledIPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function Com_IpduGroupControl.		
Multiplicity	0..*		
Type	Reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.53 BswMPduRouterControl

SWS Item	**BswM0853_Conf :**
Container Name	BswMPduRouterControl
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	

SWS Item	**BswM0854_Conf :**
Name	BswPduRouterAction

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	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: Instance		

SWS Item	**BswM0855_Conf :**		
Name	BswMPduRoutingPathGroupRef		
Description	This is a reference to the PDU Routing Path Group for which the routing in the PDU Router should be enabled or disabled. This reference corresponds to the parameter "id" of the function PduR_EnableRouting and PduR_DisableRouting.		
Multiplicity	0..*		
Type	Reference to [PduRRoutingPathGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.54 BswMTriggerStartUpPhase2

SWS Item	**BswM0920_Conf :**
Container Name	BswMTriggerStartUpPhase2
Description	This container includes all parameters needed to start phase two on a slave core. This choice shall only be chosen if multicore is used.
Configuration Parameters	

SWS Item	**BswM0921_Conf :**		
Name	BswMCoreId		
Description	This parameter defines the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.55 BswMTriggerSlaveRTEStop

SWS Item	**BswM0919_Conf :**
Container Name	BswMTriggerSlaveRTEStop
Description	This container includes all parameters needed to stop the RTE on a slave core. This choice shall only be chosen if multicore is used.
Configuration Parameters	

SWS Item	**BswM0921_Conf :**		
Name	BswMCoreId		
Description	This parameter defines the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.56 BswMUserCallout

SWS Item	**BswM0834_Conf :**
Container Name	BswMUserCallout
Description	This container includes all details needed for a user defined function call.
Configuration Parameters	

SWS Item	**BswM0843_Conf :**
Name	BswMUserCalloutFunction
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. Example usage can be: Actions to initialize other BSW modules Action to call Rte_Start() Action to call Rte_Stop() Action to call NvM_ReadAll() Action to call

	NvM_WriteAll()		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.57 BswMRteSwitch

SWS Item	**BswM0803_Conf :**
Container Name	BswMRteSwitch
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	

SWS Item	**BswM0952_Conf :**		
Name	BswMRteSwitchPortRef		
Description	This is a reference to the BswMSwitchPort.		
Multiplicity	1		
Type	Reference to [BswMSwitchPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0896_Conf :**		
Name	BswMSwitchedMode		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.58 BswMSchMSwitch

SWS Item	**BswM0899_Conf :**
Container Name	BswMSchMSwitch
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	

SWS Item	**BswM0953_Conf :**		
Name	BswMSchMSwitchPortRef		
Description	This is a reference to the BswMSwitchPort.		
Multiplicity	1		
Type	Reference to [BswMSwitchPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM0901_Conf :**		
Name	BswMSchMSwitchedMode		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.2.59 BswMSwitchIPduMode

SWS Item	**BswM00958_Conf :**
Container Name	BswMSwitchIPduMode
Description	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.
Configuration Parameters	

SWS Item	**BswM00960_Conf :**		
Name	BswMSwitchIPduModeValue		
Description	This parameter defines which transmission mode shall be selected during this call. This parameter corresponds to the parameter "truefalsemode" of the function Com_SwitchIpduTxMode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME

	Post-build time	--	
Scope / Dependency			

SWS Item	**BswM00959_Conf :**		
Name	BswMSwitchIPduModeRef		
Description	This is a reference to an I-PDU for which the transmission mode shall be set. This reference corresponds to the parameter "PduId" of the function Com_SwitchIpduTxMode.		
Multiplicity	1		
Type	Reference to [ComIPdu]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.3 Published Information

[BswM0146] 「The standardized common published parameters as required by BSW00402 in the General Requirements on Basic Software Modules [3] shall be published within the header file of this module and need to be provided in the BSW Module Description. The according module abbreviation can be found in the List of Basic Software Modules [1].」()

Additional module-specific published parameters are listed below if applicable.

11 Not applicable requirements

[BswM9999] 「 These requirements are not applicable to this specification. 」 (BSW00405, BSW170, BSW00387, BSW00399, BSW00400, BSW00336, BSW00339, BSW00409)