Document Title	Specification of Basic Software Mode Manager
Document Category	Software specification (SWS)
Document UID	313
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Removed support for some RTE lifecycle APIs (e.g. BswMRteStart, BswMPartitionRestarted, BswMCoreHaltMode ...) • New diagram showing BswM Service Ports • Editorial Changes
2025-11-27	R25-11	AUTOSAR Release Management	• Reformulated requirements related to BswMDevErrorDetect • Remodeled/Renamed BswM_ClientModeRequestEnum and BswM_OfferModeEnum as types • Editorial Changes
2024-11-27	R24-11	AUTOSAR Release Management	• Added new BswM Actions for BswMIdsMTransmissionStateChangRequest and BswMIdsMBlockStateChangeRequest • Editorial Changes
2023-11-23	R23-11	AUTOSAR Release Management	• Added BswM support for Service Discovery Control for Application Software • Added new action to set firewall state • Editorial Changes
2022-11-24	R22-11	AUTOSAR Release Management	• Removal of obsolete event request BswMWdgMRequestPartitionReset • Editorial Changes

△

2021-11-25	R21-11	AUTOSAR Release Management	• Adding new mode request for SoAd (BswM_SoAd_SoConModeChg) • Setting BswM_WdgM_RequestPartitionReset and its configuration to obsolete • Editorial Changes
2020-11-30	R20-11	AUTOSAR Release Management	• Added new notifications BswMNmStateChangeNotification and BswMSwitchAckNotification; Added new action BswMEthIfStartAllPorts • Added support for Ethernet switch port group switching • Removed support for pretended networking • Reinstated the EcuM<->BswM APIs: BswM_EcuM_CurrentState and BswM_EcuM_RequestedState
2019-11-28	R19-11	AUTOSAR Release Management	• Introduced action list priority parameter • Added configuration options for enabling/disabling SdServiceGroups • Removal of obsolete APIs: BswM_EcuM_CurrentState and BswM_EcuM_RequestedState • Editorial Changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Reworked handling of EcuM API for shutdown (e.g. BswMEcuMGoDownHaltPoll) • Removed dependency to EcuM-fixed • Changes to BswM_NvM_CurrentJobMode, BswM_ModeType, BswM_UserType and Det error codes

▽

△

2017-12-08	4.3.1	AUTOSAR Release Management	• Rework of BswM-Com interaction for Pdu group switching and deadline monitoring control. BswM-internal group vectors no longer used • BswMNmIfCarWakeUpIndication reclassified as event request port • Added new dedicated actions for Rte Start/Stop • Editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Added some actions/indications to allow for more BswM interaction with following BSW Modules: EthIf, EcuM • Waiting functionality added using the BswMTimer mode request source • Some mode requests are now modeled using BswMEventRequestPort, instead of BswMModeRequestPort • Editorial changes, increased requirement traceability and minor changes to configuration containers/parameters
2015-07-31	4.2.2	AUTOSAR Release Management	• Improved specification of service interfaces • Additional functional requirements for BswMPduGroupSwitch action • Added BswMNmIfCarWakeUpIndication as a new BswMModeRequestSource • Deprecated some spec. elements (marked with "obsolete"), editorial changes, increased requirement traceability and minor changes to configuration containers/parameters

▽

△

2014-10-31	4.2.1	AUTOSAR Release Management	• New API and configuration containers to support EcuM Fixed for Multi Core • Addition of new container for defining mode values: BswMCompuScaleModeValue • New Action BswMFrSMAllSlots for invoking FrSM_AllSlots • New requirements for: Action list execution (SWS_BswM_00223) and Deadline Monitoring (SWS_BswM_00224,00225)
2014-03-31	4.1.3	AUTOSAR Release Management	• Removal of several unnecessary parameter range checks in APIs • J1939 fix: added missing action, missing included header files • Corrections in Figures 1, 2, 3, 5 and 6 • Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Removal of several requirements from Pretended Networking chapter • Addition of new configuration parameters to several Sd related Bswm Actions • Addition of new BswM Mode Request: BswMCanSMIcomIndication • Addition of new BswM Action: BswMRteModeRequest • Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Extended to support Pretended Networking mode handling • Extended to support Heavy Duty Vehicles and J1939
2011-12-22	4.0.3	AUTOSAR Administration	• Support of Mode Machine Instances assigned to the SchM • Include of user defined header files • Possibility to provide an initial value for a BswMModeRequestPort

▽

△

2009-12-18	4.0.1	AUTOSAR Administration	• Include file BswMUserCallout.h added. This user defined header file contains declarations of the call out functions. • 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
2010-02-02	3.1.4	AUTOSAR Administration	• Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and functional overview 13
2 Acronyms and abbreviations 14
3 Related documentation 15
3.1 Input documents & related standards and norms . . . 15
3.2 Related specification . . . 15
4 Constraints and assumptions 16
4.1 Limitations . . . 16
4.2 Applicability to car domains . . . 16
5 Dependencies to other modules 17
5.1 RTE . . . 17
5.2 EcuM - Flex . . . 17
5.3 ComM . . . 17
5.4 COM . . . 17
5.5 PduR . . . 17
5.6 CanSM . . . 18
5.7 LinSM . . . 18
5.8 LinTp . . . 18
5.9 FrSM . . . 18
5.10 EthSM . . . 18
5.11 DCM . . . 18
5.12 J1939Dcm . . . 19
5.13 J1939Nm . . . 19
5.14 J1939Rm . . . 19
5.15 NM Interface . . . 19
5.16 NvM . . . 19
5.17 OS . . . 19
5.18 Sd . . . 19
5.19 File structure . . . 20
6 Requirements Tracing 21
7 Functional specification 24
7.1 Mode Arbitration . . . 24
7.1.1 Arbitration Rules . . . 24
7.1.2 Mode Conditions and Logical Expressions. . . . 25
7.1.3 Requirements of Mode Arbitration . . . 26
7.1.3.1 Immediate and Deferred Operation . . . 28
7.1.4 Arbitration Behavior after Initialization . . . 30
7.2 Mode Control . . . 31
7.2.1 Mode Processing Cycle . . . 32

7.2.2 Requirements on Mode Control . . . 34
7.2.3 Triggered and Conditional action lists . . . 36
7.2.4 Available Actions . . . 37
7.2.5 Behavior of Mode Control after Initialization . . . 37
7.3 Waiting Functionality . . . 38
7.4 Multi Partition Support . . . 39
7.5 Error classification . . . 40
7.5.1 Development Errors . . . 40
7.5.2 Runtime Errors . . . 41
7.5.3 Production Errors . . . 42
7.6 BswM Interfaces and Ports . . . 42
7.6.1 Mode Request Ports . . . 45
7.6.2 Mode Switch Ports . . . 46
7.6.3 Notifications of Mode Switches . . . 46
7.6.4 Component Type and Internal Behavior . . . 47
7.7 Ethernet switch port group switching . . . 48
7.8 Service Discovery Control . . . 48
7.8.1 General . . . 48
7.8.2 Client Service/Event Subscription Request . . . 49
7.8.3 Client Service/Event Subscription state notification . . . 52
7.8.4 Server Service Offer Request . . . 54
7.8.5 Server Service/Event Subscription State Notification . . . 56
7.8.6 Connecting ports at configuration time . . . 57
8 API specification 58
8.1 Imported types . . . 58
8.2 Type definitions . . . 59
8.2.1 BswM_ConfigType . . . 59
8.2.2 BswM_ModeType . . . 60
8.2.3 BswM_UserType . . . 60
8.3 Function definitions . . . 60
8.3.1 BswM_CanSM_CurrentState . . . 61
8.3.2 BswM_ComM_CurrentMode . . . 61
8.3.3 BswM_ComM_CurrentPNCMode . . . 62
8.3.4 BswM_ComM_InitiateReset . . . 62
8.3.5 BswM_Dcm_ApplicationUpdated . . . 63
8.3.6 BswM_Dcm_CommunicationMode_CurrentState . . . 63
8.3.7 BswM_Deinit . . . 64
8.3.8 BswM_EcuM_CurrentState . . . 65
8.3.9 BswM_EcuM_CurrentWakeup . . . 65
8.3.10 BswM_EcuM_RequestedState . . . 66
8.3.11 BswM_EthIf_PortGroupLinkStateChg . . . 66
8.3.12 BswM_EthSM_CurrentState . . . 67
8.3.13 BswM_FrSM_CurrentState . . . 67

8.3.14 BswM_Init . . . 68
8.3.15 BswM_J1939DcmBroadcastStatus . . . 69
8.3.16 BswM_J1939Nm_StateChangeNotification . . . 70
8.3.17 BswM_LinSM_CurrentSchedule . . . 70
8.3.18 BswM_LinSM_CurrentState . . . 71
8.3.19 BswM_LinTp_RequestMode . . . 71
8.3.20 BswM_Mka_ControlledPortStatus . . . 72
8.3.21 BswM_Nm_CarWakeUpIndication . . . 72
8.3.22 BswM_Nm_StateChangeNotification . . . 73
8.3.23 BswM_NvM_CurrentBlockMode . . . 74
8.3.24 BswM_NvM_CurrentJobMode . . . 74
8.3.25 BswM_RequestMode . . . 75
8.3.26 BswM_Sd_ClientServiceCurrentState . . . 75
8.3.27 BswM_Sd_ConsumedEventGroupCurrentState . . . 76
8.3.28 BswM_Sd_EventHandlerCurrentState . . . 76
8.3.29 BswM_SoAd_SoConModeChg . . . 77
8.4 Scheduled functions . . . 77
8.4.1 BswM_MainFunction . . . 78
8.5 Expected Interfaces . . . 78
8.5.1 Mandatory Interfaces . . . 78
8.5.2 Optional Interfaces . . . 79
8.6 Service Interfaces . . . 81
8.6.1 Scope of this Chapter . . . 82
8.6.2 Ports . . . 82
8.6.2.1 BswM_modeNotificationPort . . . 82
8.6.2.2 BswM_modeRequestPort . . . 82
8.6.2.3 BswM_modeSwitchPort . . . 83
8.6.2.4 S/R Port BswM_ClientServiceRequest . . . 83
8.6.2.5 S/R Port BswM_ServerServiceRequest . . . 83
8.6.2.6 Mode Switch Port BswM_ClientServiceSubscriptionState . . . 84
8.6.2.7 Mode Switch Port BswM_ServerServiceSubscriptionState . . . 84
8.6.3 Mode Declaration Groups . . . 84
8.6.3.1 ModeDeclarationGroup MDG_ClientServiceSubscriptionState . . . 84
8.6.3.2 ModeDeclarationGroup MDG_ServerServiceSubscriptionState . . . 85
8.6.4 ModeSwitchInterfaces . . . 85
8.6.4.1 ModeSwitchInterface ClientServiceSubscriptionState . . . 85
8.6.4.2 ModeSwitchInterface ServerServiceSubscriptionState . . . 86
8.6.5 S/R Interfaces . . . 86
8.6.5.1 S/R interface ClientServiceRequest . . . 86
8.6.5.2 S/R interface ServerServiceRequest . . . 87
8.6.6 Types . . . 87
8.6.6.1 BswM_ClientModeRequestType . . . 87
8.6.6.2 BswM_OfferModeType . . . 88

8.7 API to Request Port Mappings 88
9 Sequence diagrams 90
9.1 Deferred operation of BswM 90
9.2 Immediate operation of BswM 91
10 Configuration specification 92
10.1 How to read this chapter 92
10.2 Containers and configuration parameters 92
10.2.1 BswM 92
10.2.2 BswMConfig 93
10.2.3 BswMArbitration 94
10.2.4 BswMLogicalExpression 94
10.2.5 BswMModeCondition 97
10.2.6 BswMConditionValue 98
10.2.7 BswMBswMode 99
10.2.8 BswMModeDeclaration 100
10.2.9 BswMEventRequestPort 100
10.2.10 BswMModeRequestPort 101
10.2.11 BswMModeInitValue 102
10.2.12 BswMCompuScaleModeValue 103
10.2.13 BswMEventRequestSource 105
10.2.14 BswMModeRequestSource 105
10.2.15 BswMBswModeNotification 107
10.2.16 BswMCanSMIndication 108
10.2.17 BswMComMIndication 108
10.2.18 BswMComMInitiateReset 109
10.2.19 BswMComMPncRequest 110
10.2.20 BswMDcmApplicationUpdatedIndication 110
10.2.21 BswMDcmComModeRequest 111
10.2.22 BswMEcuMIndication 112
10.2.23 BswMEcuMRUNRequestIndication 112
10.2.24 BswMEcuMWakeupSource 113
10.2.25 BswMEthIfPortGroupLinkStateChg 114
10.2.26 BswMEthSMIndication 114
10.2.27 BswMFrSMIndication 115
10.2.28 BswMGenericRequest 116
10.2.29 BswMJ1939DcmBroadcastStatus 117
10.2.30 BswMJ1939NmIndication 118
10.2.31 BswMLinSMIndication 119
10.2.32 BswMLinScheduleIndication 119
10.2.33 BswMLinTpModeRequest 120
10.2.34 BswMMkaControlledPortStatusIndication 121
10.2.35 BswMNmCarWakeUpIndication 121

10.2.36 BswMNmStateChangeNotification . . . 122
10.2.37 BswMNvMJobModeIndication . . . 123
10.2.38 BswMNvMRequest . . . 124
10.2.39 BswMSdClientServiceCurrentState . . . 125
10.2.40 BswMSdConsumedEventGroupCurrentState . . . 125
10.2.41 BswMSdEventHandlerCurrentState . . . 126
10.2.42 BswMSoAdSoConModeChg . . . 127
10.2.43 BswMSwcModeNotification . . . 128
10.2.44 BswMSwcModeRequest . . . 128
10.2.45 BswMSwitchAckNotification . . . 129
10.2.46 BswMTimer . . . 130
10.2.47 BswMRule . . . 131
10.2.48 BswMDataTypeMappingSets . . . 133
10.2.49 BswMModeControl . . . 134
10.2.50 BswMAction . . . 134
10.2.51 BswMAvailableActions . . . 135
10.2.52 BswMClearEventRequest . . . 137
10.2.53 BswMComMAllowCom . . . 138
10.2.54 BswMComMModeLimitation . . . 139
10.2.55 BswMComMModeSwitch . . . 140
10.2.56 BswMDeadlineMonitoringControl . . . 141
10.2.57 BswMEcuMDriverInitListBswM . . . 143
10.2.58 BswMEcuMGoDownHaltPoll . . . 144
10.2.59 BswMEcuMSelectShutdownTarget . . . 145
10.2.60 BswMEcuMStateSwitch . . . 146
10.2.61 BswMEthIfStartAllPorts . . . 147
10.2.62 BswMEthIfSwitchPortGroupRequestMode . . . 148
10.2.63 BswMFrSMAllSlots . . . 149
10.2.64 BswMIdsMBlockStateChangeRequest . . . 150
10.2.65 BswMIdsMTransmissionStateChangeRequest . . . 150
10.2.66 BswMJ1939DcmStateSwitch . . . 151
10.2.67 BswMJ1939RmStateSwitch . . . 153
10.2.68 BswMLinScheduleSwitch . . . 154
10.2.69 BswMNMControl . . . 155
10.2.70 BswMPduGroupSwitch . . . 156
10.2.71 BswMPduRouterControl . . . 158
10.2.72 BswMRteModeRequest . . . 160
10.2.73 BswMRteSwitch . . . 161
10.2.74 BswMSchMSwitch . . . 162
10.2.75 BswMSdClientServiceModeRequest . . . 163
10.2.76 BswMSdConsumedEventGroupModeRequest . . . 164
10.2.77 BswMSdServerServiceModeRequest . . . 165
10.2.78 BswMSdServiceGroupSwitch . . . 166

10.2.79 BswMSetFirewallState . . . 168
10.2.80 BswMSwitchIPduMode . . . 168
10.2.81 BswMTimerControl . . . 170
10.2.82 BswMTriggerIPduSend . . . 171
10.2.83 BswMUserCallout . . . 172
10.2.84 BswMActionList . . . 173
10.2.85 BswMActionListItem . . . 174
10.2.86 BswMRteModeRequestPort . . . 176
10.2.87 BswMSwitchPort . . . 177
10.2.88 BswMGeneral . . . 178
10.2.89 BswMUserIncludeFiles . . . 185
10.3 Published Information . . . 186
A Not applicable requirements 187
B Change history of AUTOSAR traceable items 188
B.1 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 188
B.1.1 Added Specification Items in R23-11 . . . 188
B.1.2 Changed Specification Items in R23-11 . . . 189
B.1.3 Deleted Specification Items in R23-11 . . . 189
B.1.4 Added Constraints in R23-11 . . . 190
B.1.5 Changed Constraints in R23-11 . . . 190
B.1.6 Deleted Constraints in R23-11 . . . 190
B.1.7 Added Specification Items in R24-11 . . . 190
B.1.8 Changed Specification Items in R24-11 . . . 190
B.1.9 Deleted Specification Items in R24-11 . . . 192
B.1.10 Added Constraints in R24-11 . . . 193
B.1.11 Changed Constraints in R24-11 . . . 193
B.1.12 Deleted Constraints in R24-11 . . . 193
B.1.13 Added Specification Items in R25-11 . . . 193
B.1.14 Changed Specification Items in R25-11 . . . 194
B.1.15 Deleted Specification Items in R25-11 . . . 209
B.1.16 Added Constraints in R25-11 . . . 210
B.1.17 Changed Constraints in R25-11 . . . 210
B.1.18 Deleted Constraints in R25-11 . . . 210
B.1.19 Added Specification Items in R26-11 . . . 210
B.1.20 Changed Specification Items in R26-11 . . . 211
B.1.21 Deleted Specification Items in R26-11 . . . 221
B.1.22 Added Constraints in R26-11 . . . 222
B.1.23 Changed Constraints in R26-11 . . . 222
B.1.24 Deleted Constraints in R26-11 . . . 222

1 Introduction and functional overview

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

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

2 Acronyms and abbreviations

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

Table 2.1: Table of acronyms and abbreviations

3 Related documentation

3.1 Input documents & related standards and norms

[1] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[2] Guide to Mode Management
AUTOSAR_CP_EXP_ModeManagementGuide

[3] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[4] Requirements on Mode Management
AUTOSAR_CP_RS_ModeManagement

[5] Specification of RTE Software
AUTOSAR_CP_SWS_RTE

[6] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

3.2 Related specification

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

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

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

4 Constraints and assumptions

4.1 Limitations

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

4.2 Applicability to car domains

The BSW Mode Manager is applicable to all car domains.

5 Dependencies to other modules

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

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

5.1 RTE

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

5.2 EcuM - Flex

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

5.3 ComM

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

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

5.4 COM

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

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

BswM can also trigger transmission of an I-PDU.

5.5 PduR

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

5.6 CanSM

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

5.7 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.

5.8 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.

5.9 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`.

5.10 EthSM

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

5.11 DCM

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

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

5.12 J1939Dcm

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

5.13 J1939Nm

The J1939Nm provides a state indication via `BswM_J1939Nm_StateChangeNotification`.

5.14 J1939Rm

BswM changes states of J1939Rm via `J1939Rm_SetState`.

5.15 NM Interface

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

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

The Nm uses `BswM_Nm_CarWakeUpIndication` to indicate a CarWakeup.

5.16 NvM

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

5.17 OS

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

5.18 Sd

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

5.19 File structure

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

6 Requirements Tracing

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

Requirement	Description	Satisfied by
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_BswM_00002] [SWS_BswM_00043] [SWS_BswM_00044] [SWS_BswM_00261]
[SRS_BSW_00167]	All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks	[SWS_BswM_00240] [SWS_BswM_00242] [SWS_BswM_00243] [SWS_BswM_00256]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_BswM_00001] [SWS_BswM_00237] [SWS_BswM_00279]
[SRS_BSW_00305]	Data types naming convention	[SWS_BswM_00041]
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_BswM_00321] [SWS_BswM_00322] [SWS_BswM_00323] [SWS_BswM_00324]
[SRS_BSW_00336]	Basic SW module shall be able to shutdown	[SWS_BswM_00119] [SWS_BswM_00120]
[SRS_BSW_00344]	BSW Modules shall support link-time configuration	[SWS_BswM_00002]
[SRS_BSW_00358]	The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void	[SWS_BswM_00002]
[SRS_BSW_00373]	The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention	[SWS_BswM_00053]
[SRS_BSW_00384]	The Basic Software Module specifications shall specify at least in the description which other modules they require	[SWS_BswM_00007] [SWS_BswM_00008]
[SRS_BSW_00385]	List possible error notifications	[SWS_BswM_00230]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_BswM_00002] [SWS_BswM_00042] [SWS_BswM_00213]
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_BswM_00002]
[SRS_BSW_00414]	Init functions shall have a pointer to a configuration structure as single parameter	[SWS_BswM_00002]
[SRS_BSW_00441]	Naming convention for type, macro and function	[SWS_BswM_00213] [SWS_BswM_00214] [SWS_BswM_00216]
[SRS_BSW_00452]	Classification of runtime errors	[SWS_BswM_00238] [SWS_BswM_00239]
[SRS_BSW_00467]	The init / deinit services shall only be called by BswM or EcuM	[SWS_BswM_00118]



△

Requirement	Description	Satisfied by
[SRS_ModeMgm_-02000]	Mode Arbitration on Request	[SWS_BswM_00009] [SWS_BswM_00013] [SWS_BswM_00014] [SWS_BswM_00015] [SWS_BswM_00016] [SWS_BswM_00019] [SWS_BswM_00035] [SWS_BswM_00059] [SWS_BswM_00060] [SWS_BswM_00064] [SWS_BswM_00066] [SWS_BswM_00068] [SWS_BswM_00069] [SWS_BswM_00075] [SWS_BswM_00189] [SWS_BswM_00203] [SWS_BswM_00220] [SWS_BswM_00241] [SWS_BswM_00251] [SWS_BswM_00252] [SWS_BswM_00253] [SWS_BswM_00257] [SWS_BswM_00258] [SWS_BswM_00264] [SWS_BswM_00278] [SWS_BswM_00281] [SWS_BswM_00282] [SWS_BswM_00325] [SWS_BswM_00326]
[SRS_ModeMgm_-02001]	Configuration of Mode Requesters	[SWS_BswM_00013] [SWS_BswM_00014] [SWS_BswM_00046] [SWS_BswM_00047] [SWS_BswM_00048] [SWS_BswM_00049] [SWS_BswM_00050] [SWS_BswM_00051] [SWS_BswM_00052] [SWS_BswM_00058] [SWS_BswM_00059] [SWS_BswM_00104] [SWS_BswM_00131] [SWS_BswM_00148] [SWS_BswM_00152] [SWS_BswM_00156] [SWS_BswM_00158] [SWS_BswM_00165] [SWS_BswM_00194] [SWS_BswM_00200] [SWS_BswM_00201] [SWS_BswM_00203] [SWS_BswM_00216] [SWS_BswM_00217] [SWS_BswM_00220] [SWS_BswM_00235] [SWS_BswM_00236] [SWS_BswM_00249] [SWS_BswM_00250] [SWS_BswM_00254] [SWS_BswM_00256] [SWS_BswM_00257] [SWS_BswM_00261] [SWS_BswM_00263] [SWS_BswM_00283] [SWS_BswM_00284] [SWS_BswM_00285] [SWS_BswM_91001] [SWS_BswM_91002] [SWS_BswM_91003] [SWS_BswM_91004]
[SRS_ModeMgm_-02002]	Configuration of Mode Arbitration Rules	[SWS_BswM_00010] [SWS_BswM_00012] [SWS_BswM_00060] [SWS_BswM_00061] [SWS_BswM_00064] [SWS_BswM_00068] [SWS_BswM_00069] [SWS_BswM_00075] [SWS_BswM_00117] [SWS_BswM_00203] [SWS_BswM_00241] [SWS_BswM_00242] [SWS_BswM_00243] [SWS_BswM_00244] [SWS_BswM_00245] [SWS_BswM_00246] [SWS_BswM_00247] [SWS_BswM_00248] [SWS_BswM_00252] [SWS_BswM_00253] [SWS_BswM_00256] [SWS_BswM_00258] [SWS_BswM_00281] [SWS_BswM_00282]

▽

△

Requirement	Description	Satisfied by
[SRS_ModeMgm_-02003]	Performing Mode Specific Actions	[SWS_BswM_00011] [SWS_BswM_00015] [SWS_BswM_00016] [SWS_BswM_00017] [SWS_BswM_00018] [SWS_BswM_00019] [SWS_BswM_00023] [SWS_BswM_00037] [SWS_BswM_00038] [SWS_BswM_00039] [SWS_BswM_00040] [SWS_BswM_00054] [SWS_BswM_00055] [SWS_BswM_00062] [SWS_BswM_00066] [SWS_BswM_00067] [SWS_BswM_00115] [SWS_BswM_00116] [SWS_BswM_00147] [SWS_BswM_00214] [SWS_BswM_00223] [SWS_BswM_00255] [SWS_BswM_00260] [SWS_BswM_00262] [SWS_BswM_00265] [SWS_BswM_00269] [SWS_BswM_00273] [SWS_BswM_00274] [SWS_BswM_00275] [SWS_BswM_00276] [SWS_BswM_00277] [SWS_BswM_00320]
[SRS_ModeMgm_-02004]	Mode Notification to Users	[SWS_BswM_00202]
[SRS_ModeMgm_-02005]	Initialize BSW	[SWS_BswM_00131] [SWS_BswM_00269] [SWS_BswM_91003] [SWS_BswM_91004]
[SRS_ModeMgm_-02006]	Prepare BSW for Shutdown	[SWS_BswM_00131] [SWS_BswM_91003] [SWS_BswM_91004]
[SRS_ModeMgm_-02007]	Set Communication Modes	[SWS_BswM_00273] [SWS_BswM_00274] [SWS_BswM_00286]
[SRS_ModeMgm_-02008]	Control Service Discovery	[SWS_BSWM_91006] [SWS_BSWM_91007] [SWS_BSWM_91008] [SWS_BSWM_91009] [SWS_BSWM_91010] [SWS_BSWM_91011] [SWS_BSWM_91012] [SWS_BSWM_91013] [SWS_BSWM_91014] [SWS_BSWM_91015] [SWS_BSWM_91016] [SWS_BSWM_91018] [SWS_BswM_00204] [SWS_BswM_00207] [SWS_BswM_00210] [SWS_BswM_00280] [SWS_BswM_00287] [SWS_BswM_00288] [SWS_BswM_00289] [SWS_BswM_00290] [SWS_BswM_00291] [SWS_BswM_00292] [SWS_BswM_00293] [SWS_BswM_00294] [SWS_BswM_00295] [SWS_BswM_00296] [SWS_BswM_00297] [SWS_BswM_00298] [SWS_BswM_00299] [SWS_BswM_00300] [SWS_BswM_00301] [SWS_BswM_00302] [SWS_BswM_00303] [SWS_BswM_00304] [SWS_BswM_00305] [SWS_BswM_00306] [SWS_BswM_00307] [SWS_BswM_00308] [SWS_BswM_00309] [SWS_BswM_00310] [SWS_BswM_00311] [SWS_BswM_00312] [SWS_BswM_00313] [SWS_BswM_00314] [SWS_BswM_00315] [SWS_BswM_00316] [SWS_BswM_00317] [SWS_BswM_00318] [SWS_BswM_00319] [SWS_BswM_91005]

Table 6.1: Requirements Tracing

7 Functional specification

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

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

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

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

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

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

7.1 Mode Arbitration

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

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

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

7.1.1 Arbitration Rules

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

7.1.2 Mode Conditions and Logical Expressions.

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



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

[SWS_BswM_00252]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈When a BswMModeCondition has BswMConditionType = BSWM_EVENT_IS_SET and references a BswMEventRequestPort:

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

⌋

[SWS_BswM_00253]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈When a BswMModeCondition has BswMConditionType = BSWM_EVENT_IS_CLEARED and references a BswMEventRequestPort:

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

⌋

[SWS_BswM_00254]

Upstream requirements: SRS_ModeMgm_02001

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

[SWS_BswM_00255]

Upstream requirements: SRS_ModeMgm_02003

⌈When a BswMClearEventRequest action is executed on a BswMEventRequestPort, the BswMEventRequestPort shall be placed in a CLEAR state.⌋

7.1.3 Requirements of Mode Arbitration

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

[SWS_BswM_00009]

Upstream requirements: SRS_ModeMgm_02000

⌈The BswM shall perform mode arbitration based on incoming mode requests.⌋

[SWS_BswM_00035]

Upstream requirements: SRS_ModeMgm_02000

⌈The BswM shall perform mode arbitration based on incoming mode indications.⌋

[SWS_BswM_00278]

Upstream requirements: SRS_ModeMgm_02000

⌈The BswM shall perform mode arbitration based on event requests as well as the clearing of event requests.⌋

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

[SWS_BswM_00010]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall perform mode arbitration using configured rules.⌋

[SWS_BswM_00012]

Upstream requirements: SRS_ModeMgm_02002

⌈The mode arbitration rules shall be configurable using the module configuration parameters.⌋

[SWS_BswM_00117]

Upstream requirements: SRS_ModeMgm_02002

⌈BswM is not allowed to use results of previous arbitration rule evaluations as input for the logical expressions.⌋

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

[SWS_BswM_00147]

Upstream requirements: SRS_ModeMgm_02003

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

[SWS_BswM_00189]

Upstream requirements: SRS_ModeMgm_02000

⌈The BswM shall perform mode arbitration based on incoming mode switch notifications.⌋

7.1.3.1 Immediate and Deferred Operation

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

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

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

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

[SWS_BswM_00061]

Upstream requirements: SRS_ModeMgm_02002

⌈A mode arbitration rule may contain any combination of immediate and deferred mode arbitration requests.⌋

[SWS_BswM_00013]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001

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

[SWS_BswM_00059]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001

⌈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.⌋

[SWS_BswM_00014]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001

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

[SWS_BswM_00257]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001

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

[SWS_BswM_00258]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

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

[SWS_BswM_00060]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈All rules that use at least one deferred mode condition shall be evaluated during every execution of the main function of BswM.⌋

[SWS_BswM_00068]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

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

[SWS_BswM_00069]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

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

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

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

[SWS_BswM_00281]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

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

[SWS_BswM_00282]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈BswM shall update the value of a DEFERRED mode request port when the mode request port is triggered.⌋

Situations might occur where a background task could request a mode switch, and is then pe-empted by the task running the BswM main function. In such situations, the "logical" outcome may be different and depend on the actual timing and prioritization of the tasks.

7.1.4 Arbitration Behavior after Initialization

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

[SWS_BswM_00064]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

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

[SWS_BswM_00241]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈BswM shall only arbitrate rules that do not contain any undefined mode conditions within its logical expressions.⌋

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

[SWS_BswM_00203]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001, SRS_ModeMgm_02002

⌈In case `BswMModeInitValue` is defined the BswM shall initialize the corresponding `BswMModeRequestSource` with either the `BswMBswModeInitValue` or the `BswMCompuScaleModeValue` while the BswM is initialized. The BswM shall reject configurations which contain both a `BswMBswModeInitValue` and a `BswMCompuScaleModeValue` for a single `BswMModeInitValue`. This initialization value shall be used

for the arbitration rule until the corresponding mode arbitration request has been updated e.g. each call of `BswM_RequestMode` shall update the GenericRequest mode.⌋

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

[SWS_BswM_00251]

Upstream requirements: SRS_ModeMgm_02000

⌈Upon initialization of the BswM, all `BswMEventRequestPort`s shall be initialized to a CLEAR state.⌋

7.2 Mode Control

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

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

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

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

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

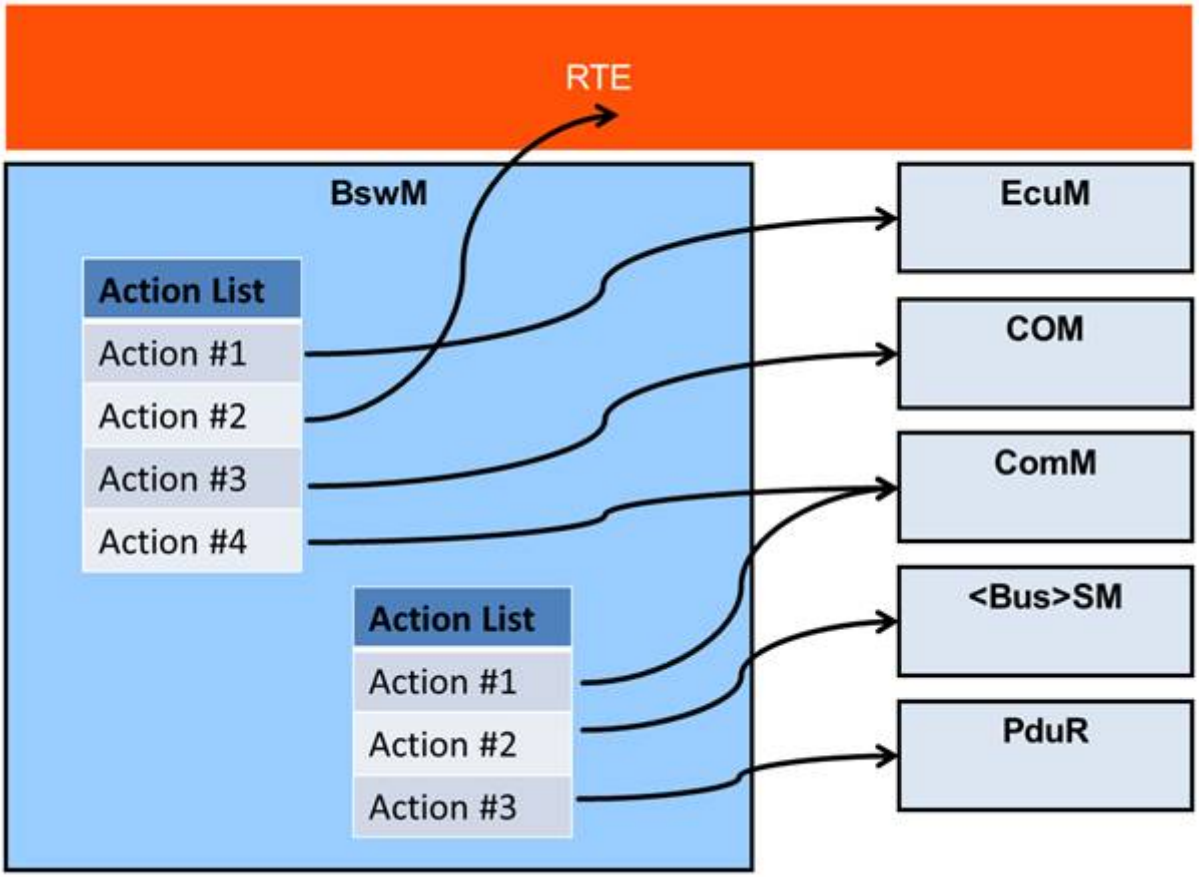


Figure 7.2: Example showing two action lists

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

7.2.1 Mode Processing Cycle

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

1. The Mode Requester SW-C requests mode A through its Sender Port. The RTE distributes the request and the BswM receives it through its Receiver Port.
2. The BswM evaluates its rules either as a result of a received mode arbitration request, or cyclically during the execution of the BswM main function.
3. The corresponding Action List is executed according to the selected execution method (see section "Triggered and Conditional action lists").

4. When executing the Action List, the BswM may issue one or several calls to the RTE Switch API [5] as actions to inform the affected SW-Cs about the arbitration result. Any SW-C, especially the mode requester can register to receive the mode switch indications.

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

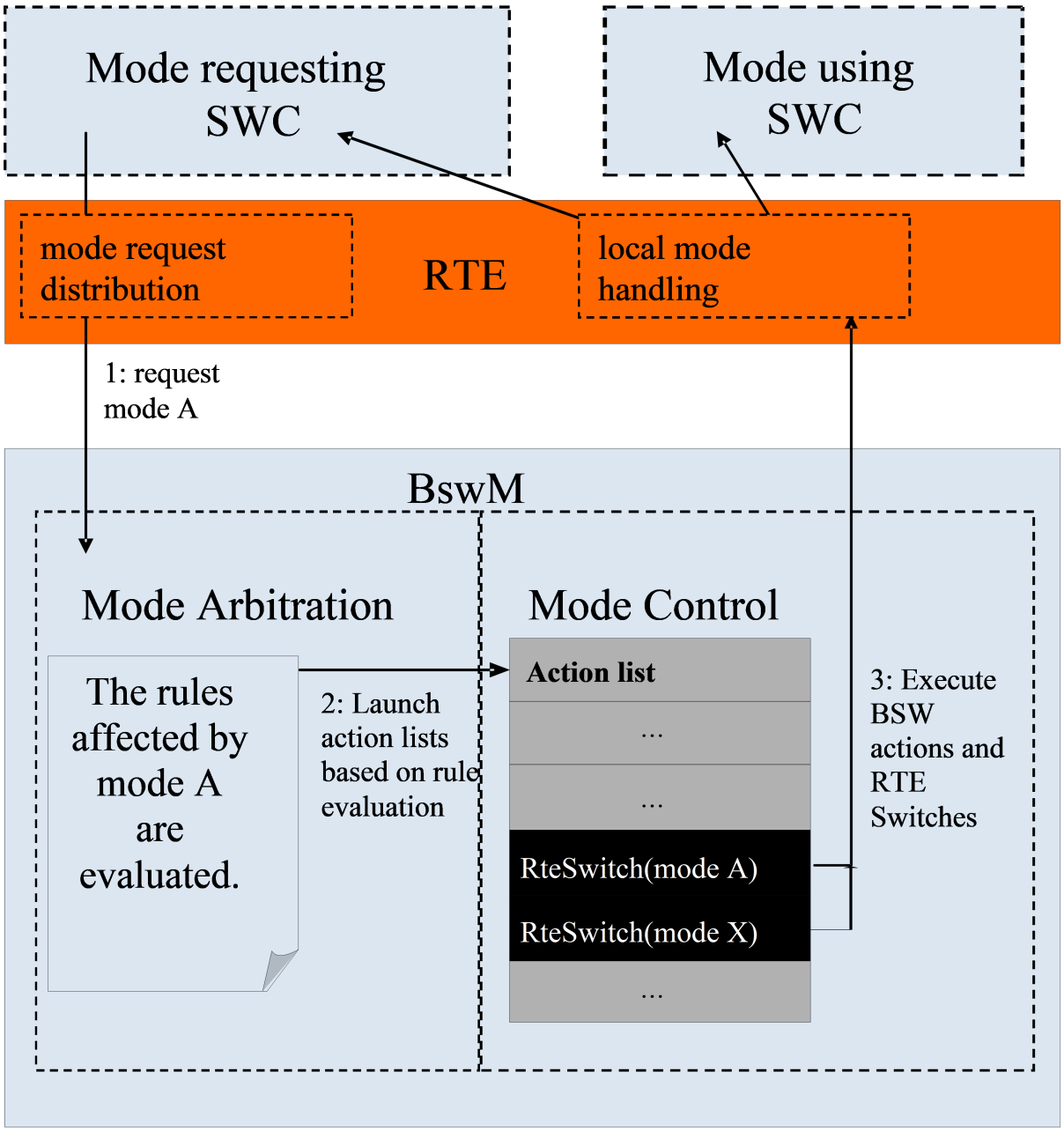


Figure 7.3: Mode Processing Cycle

7.2.2 Requirements on Mode Control

[SWS_BswM_00016]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02003

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

[SWS_BswM_00015]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02003

⌈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.⌋

[SWS_BswM_00017]

Upstream requirements: SRS_ModeMgm_02003

⌈An action list comprises a set of actions that BswM shall execute in an ordered manner.⌋

[SWS_BswM_00018]

Upstream requirements: SRS_ModeMgm_02003

⌈An action list may contain links to other action lists that BswM shall include in the execution.⌋

[SWS_BswM_00019]

Upstream requirements: SRS_ModeMgm_02003, SRS_ModeMgm_02000

⌈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.⌋

[SWS_BswM_00067]

Upstream requirements: SRS_ModeMgm_02003

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

[SWS_BswM_00037]

Upstream requirements: SRS_ModeMgm_02003

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

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

[SWS_BswM_00062]

Upstream requirements: SRS_ModeMgm_02003

⌈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.⌋

[SWS_BswM_00223]

Upstream requirements: SRS_ModeMgm_02003

⌈If a top-level action list is triggered by multiple rules during mode arbitration, this shall result in a single trigger to execute the action list during mode control.⌋

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

[SWS_BswM_00275]

Upstream requirements: SRS_ModeMgm_02003

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

[SWS_BswM_00276]

Upstream requirements: SRS_ModeMgm_02003

⌈The `BswMActionListPriority` shall be ignored for action lists which are not top-level action lists.⌋

[SWS_BswM_00277]

Upstream requirements: SRS_ModeMgm_02003

⌈An Action List with no `BswMActionListPriority` shall be interpreted as having a `BswMActionListPriority` equal to 0.⌋

[SWS_BswM_CONSTR_00001] ⌈The BswM shall reject configurations where a `BswMActionList` contains `BswMActionListItem`s with same-valued `BswMActionListItemIndex`es.⌋

[SWS_BswM_00260]

Upstream requirements: SRS_ModeMgm_02003

⌈When executing a `BswMActionList`: the BswM shall start with the `BswMActionListItem` that has the lowest-valued `BswMActionListItemIndex`. Subsequent

BswMActionListItems shall be executed in increasing order of their BswMActionListItemIndex.⌋

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

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

7.2.3 Triggered and Conditional action lists

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

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

[SWS_BswM_00011]

Upstream requirements: SRS_ModeMgm_02003

⌈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.⌋

[SWS_BswM_00023]

Upstream requirements: SRS_ModeMgm_02003

⌈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.⌋

[SWS_BswM_00115]

Upstream requirements: SRS_ModeMgm_02003

⌈If a True action list is configured for conditional execution, the BswM shall execute it every time the corresponding rule is evaluated to True.⌋

[SWS_BswM_00116]

Upstream requirements: SRS_ModeMgm_02003

⌈If a False action list is configured for conditional execution, the BswM shall execute it every time the corresponding rule is evaluated to False.⌋

[SWS_BswM_00055]

Upstream requirements: SRS_ModeMgm_02003

⌈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".

If an action consists of multiple calls (e.g. BswMTriggerIPduSend) and one of those calls returns E_NOT_OK, the remaining calls of that action shall also be aborted.⌋

7.2.4 Available Actions

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

[SWS_BswM_00038]

Upstream requirements: SRS_ModeMgm_02003

⌈BswM shall be able to execute the predefined actions defined by configuration container `BswMAvailableActions`.⌋

[SWS_BswM_00039]

Upstream requirements: SRS_ModeMgm_02003

⌈The BswM shall be able to call any function in the AUTOSAR BSW even though it is not among the standardized actions defined in `BswMAvailableActions`.⌋

[SWS_BswM_00040]

Upstream requirements: SRS_ModeMgm_02003

⌈The BswM shall be able to call user defined functions.⌋

[SWS_BswM_00054]

Upstream requirements: SRS_ModeMgm_02003

⌈The parameters of the user defined functions, and their values, shall be defined at ECU configuration time using the `BswMUserCallout` configuration container.⌋

7.2.5 Behavior of Mode Control after Initialization

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

[SWS_BswM_00066] The BswM shall act according to what is stated in this table when a rule is evaluated for the first time after initialization. Note: The "true" and "false" action lists are optional for each rule.

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02003

⌈

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

⌋

7.3 Waiting Functionality

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

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

[SWS_BswM_00261]

Upstream requirements: SRS_BSW_00101, SRS_ModeMgm_02001

⌈Each BswMTimer shall be stopped (BSWM_TIMER_STOPPED) during initialization.⌋

[SWS_BswM_00262]

Upstream requirements: SRS_ModeMgm_02003

⌈The action BswMTimerAction BSWM_TIMER_START shall reload the referenced BswMTimer (via BswMTimerRef) with the corresponding timer value (refer Bsw MTimerValue) and change the mode of the timer to BSWM_TIMER_STARTED.⌋

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

[SWS_BswM_00263]

Upstream requirements: SRS_ModeMgm_02001

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

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

[SWS_BswM_00264]

Upstream requirements: SRS_ModeMgm_02000

⌈In case a BswMTimer which is in mode BSWM_TIMER_STARTED expires, its mode shall be changed to BSWM_TIMER_EXPIRED, and then the BswMTimer mode shall be arbitrated in the same BswM_MainFunction cycle.⌋

[SWS_BswM_00265]

Upstream requirements: SRS_ModeMgm_02003

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

[SWS_BswM_00220]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02001

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

Note: A BswMTimer in mode BSWM_TIMER_EXPIRED will not automatically be set to BSWM_TIMER_STOPPED by the BswM. An action will need to be configured by the user in order to transition the BswMTimer from BSWM_TIMER_EXPIRED into another mode. If there is no action configured to transition the BswMTimer out of mode BSWM_TIMER_EXPIRED, then the BswMTimer will continue to be arbitrated as BSWM_TIMER_EXPIRED in the following BswM main function cycles.

7.4 Multi Partition Support

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

[SWS_BswM_00320] Partition Local Action Lists

Upstream requirements: SRS_ModeMgm_02003

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

Regardless, it is possible to use any mode request on all BswM instances.

[SWS_BswM_00325] Arbitration Independence

Upstream requirements: SRS_ModeMgm_02000

⌈A `BswMArbitration` shall be executed independently of which BswM instance the `BswMModeRequestPort` or `BswMEventRequestPort` is issued.⌋

[SWS_BswM_00326] Deferred Rule Arbitration for Separate BswM Instances

Upstream requirements: SRS_ModeMgm_02000

⌈In case the `BswMModeRequestPort` or `BswMEventRequestPort` is on another BswM instance than the `BswMArbitration`, the BswM shall execute the rule always in a DEFERRED manner as defined in [SWS_BswM_00060] regardless of attribute `BswMEventRequestProcessing` or `BswMRequestProcessing`.⌋

7.5 Error classification

Chapter [1, General Specification of Basic Software Modules] 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

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

7.5.1 Development Errors

[SWS_BswM_00230] Definition of development errors in module BswM

Upstream requirements: SRS_BSW_00385

⌈

Type of error	*Related error code*	*Error value*
A service was called prior to initialization	BSWM_E_UNINIT	0x01
A requesting user was out of range	BSWM_E_REQ_USER_OUT_OF_RANGE	0x04
A requested mode was out of range	BSWM_E_REQ_MODE_OUT_OF_RANGE	0x05
The provided configuration is inconsistent	BSWM_E_PARAM_CONFIG	0x06
Invalid network handle in parameter list	BSWM_E_INVALID_NETWORK_HANDLE	0x09

▽

△

Type of error	Related error code	Error value
Invalid handle in parameter list	BSWM_E_INVALID_HANDLE_ID	0x10

⌋

[SWS_BswM_00321] Development Errors for Invalid Network Handle

Upstream requirements: SRS_BSW_00323

⌈If the BswMDevErrorDetect switch is enabled, APIs provided by the BswM accepting a parameter of type NetworkHandleType or PNCHandleType, shall check if a valid handle is passed. If not, the API shall report BSWM_E_INVALID_NETWORK_HANDLE to the Default Error Tracer.⌋

[SWS_BswM_00322] Development Errors for Invalid Handle

Upstream requirements: SRS_BSW_00323

⌈If the BswMDevErrorDetect switch is enabled, APIs provided by the BswM accepting a parameter <*>HandleId, shall check if a valid HandleId is passed. If not, the API shall report BSWM_E_INVALID_HANDLE_ID to the Default Error Tracer.⌋

[SWS_BswM_00323] Development Errors for Out of Range

Upstream requirements: SRS_BSW_00323

⌈If the BswMDevErrorDetect switch is enabled, APIs provided by the BswM accepting a parameter <*>ModeType or <*>StateType , shall check if a valid mode/state is passed. If not, the API shall report BSWM_E_REQ_MODE_OUT_OF_RANGE to the Default Error Tracer.⌋

[SWS_BswM_00324] Development Errors for other Parameters

Upstream requirements: SRS_BSW_00323

⌈Unless otherwise covered by [SWS_BswM_00321], [SWS_BswM_00322], [SWS_BswM_00323]: If the BswMDevErrorDetect switch is enabled, APIs provided by the BswM accepting parameters shall check the parameters for being in the allowed range. If not, the API shall report BSWM_E_PARAM_INVALID to the Default Error Tracer.⌋

7.5.2 Runtime Errors

[SWS_BswM_00238] Definition of runtime errors in module BswM

Upstream requirements: SRS_BSW_00452

⌈

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

⌋

[SWS_BswM_00239]

Upstream requirements: SRS_BSW_00452

⌈If `BswMReportFailRuntimeErrorId` is configured for a `BswMActionListItem`, then the BswM shall report a BSWM_E_ACTION_FAILED Runtime Error to Det if the action returns E_NOT_OK. The ErrorId reported in the BSWM_E_ACTION_FAILED Runtime Error is given by the value configured in `BswMReportFailRuntime-ErrorId`.⌋

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

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

[SWS_BswM_00240]

Upstream requirements: SRS_BSW_00167

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

7.5.3 Production Errors

There are no production errors.

7.6 BswM Interfaces and Ports

This chapter specifies the AUTOSAR Interfaces and Ports that are provided by the Basic Software Mode Manager. Note that ports on both sides of the RTE are required: The SW-C description of the Basic Software Mode Manager services will define the ports below the RTE. Each AUTOSAR SW-C, which uses the services, must contain service ports in its own SW-C description. These ports are typed with the same interfaces and have to be connected to the ports of the Basic Software Mode Manager, so that the RTE can generate the appropriate IDs and the required symbols.

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

the ImplementationDataType of the data element is mapped to the ModeDeclaration Group).

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

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

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

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


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


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

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

Figure 7.4 shows how the ports of the application SW-Cs connect to the service ports of the BSW Mode Manager. The Application Mode Manager SW-C has a Mode Request Port and a Mode Switch R-port (named `modeNotificationPort` to distinguish it

from the Mode Switch P-ports). The first port is to request changes in its application mode, the latter to receive notifications when the BswM has performed the mode change. The Mode Request Port of the Application Mode Manager (`modeRequestPort0`) connects to the corresponding Mode Request Port of the BSW Mode Manager. Since this is normal Sender/Receiver communication, the Application Mode Manager may connect to multiple BSW Mode Managers even on remote ECUs.

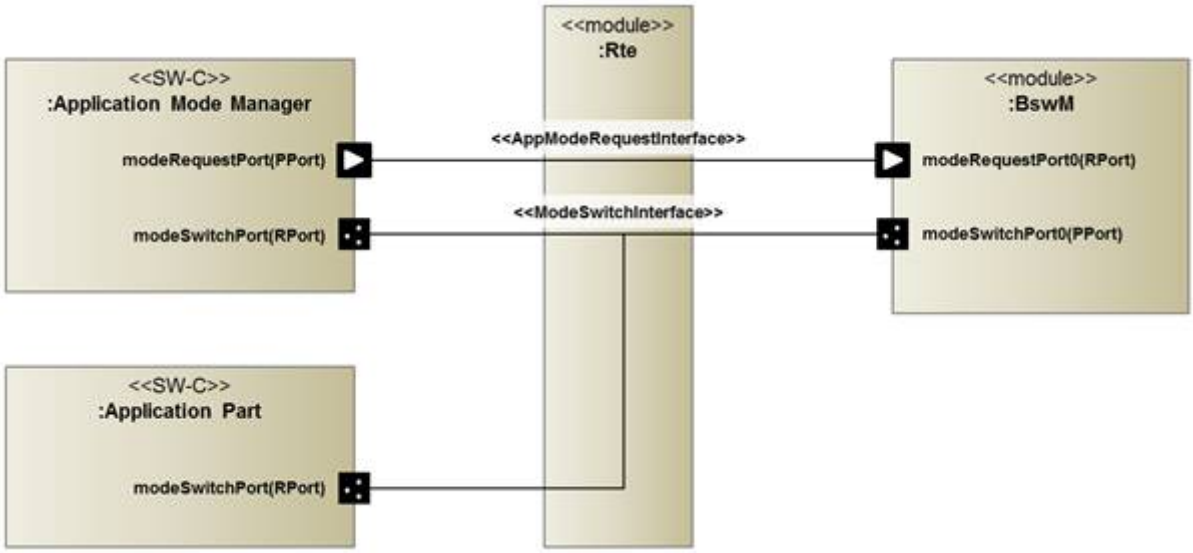


Figure 7.4: Connections between Application Mode Manager, Application Parts and the BSW Mode Manager

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

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

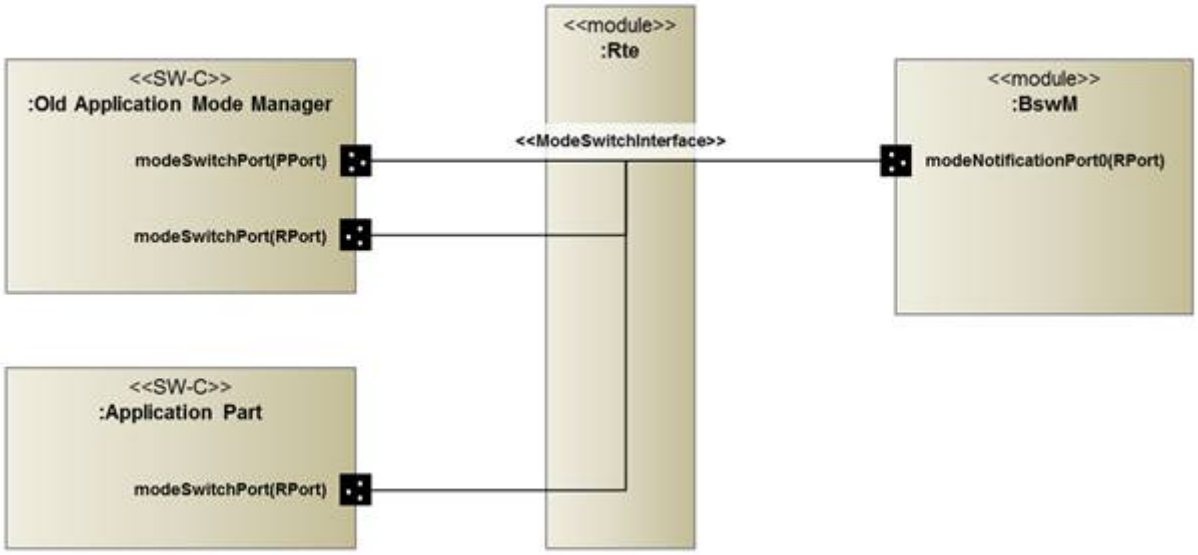


Figure 7.5: Connections between SW-C based Application Mode Manager, Application Parts and the BSW Mode Manager

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

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

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

```
ARPackage AUTOSAR_BswM/BswModuleDescription
```

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

7.6.1 Mode Request Ports

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

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

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

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

7.6.2 Mode Switch Ports

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

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

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

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

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

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

7.6.3 Notifications of Mode Switches

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

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

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

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

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

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

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

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

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

7.6.4 Component Type and Internal Behavior

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

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

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

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


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

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

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

7.7 Ethernet switch port group switching

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

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

7.8 Service Discovery Control

7.8.1 General

Besides the generic means for requesting and querying modes, the BswM provides a standardized interface to control Service Discovery by Applicative SWCs.

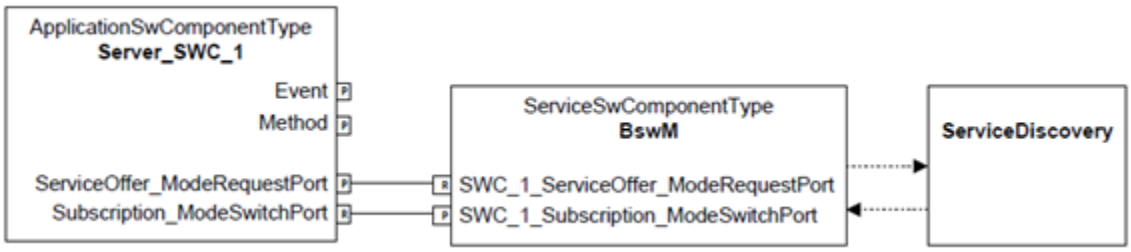


Figure 7.6: BswM interface for Service Discovery by ASWCs

Based on the input from SWC description and ECUC a configuration shall be generated as described in the following chapters.

Input for the generation are the ports with RoleBasedDataAssignment of ServerServiceOffer, ServerEventSubscriptionStatus, ClientEventSubscription and ClientEventSubscriptionStatus together with their RepresentedPortGroups.

See chapter 13.6.5 Service Oriented Use Cases in [6] for details on how to model neccessary ports and relations.

Following steps, fulfilled by configuration tools are neccessary:

1. Analyze SWC descriptions to find the affected ports

2. Analyze the ECUC to deduce the related entities in Service Discovery
3. Create Rules and ActionLists to implement the desired behavior provided by BswM
4. Create the BswMModeRequestSources for the notifications from Service Discovery Module(s)
5. Create the BswM side of the required S/R ports serving as input for the rules
6. Create the BswM side of the provided ModeSwitch ports for indications towards SWCs
7. Connect the ports between BswM Service SWC and Applications SWCs based on the well-defined naming schema

Naming patterns are used to facilitate the mapping of the ports between BswM and SWCs. They are designed in a way that a matching of (parts of the) names can be applied. By using the names of the SWCs ports where possible, a decoupling of the interface is achieved, keeping the SWCs agnostic from underlying communication details.

7.8.2 Client Service/Event Subscription Request

A software-component that implements the client functionality acts as a mode requester and exposes a PPortPrototype typed by a SenderReceiverInterface. By this means, the software-component can send mode requests towards the BswM in order to subscribe and stop subscribe of dedicated events and methods.

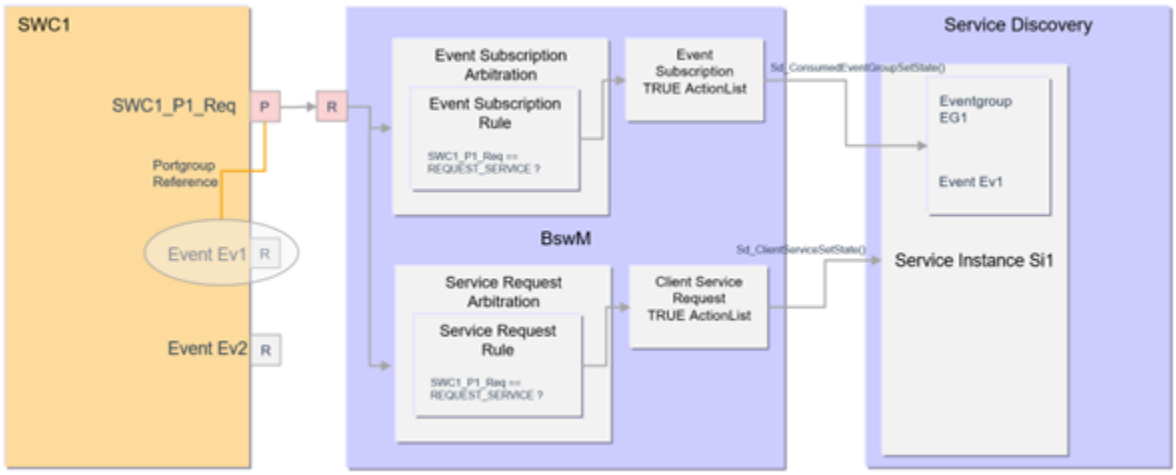


Figure 7.7: Chain of involved Entities building up an Event Subscription Request

Figure 7.7 shows the chain that builds up an event subscription request in case (SomeIP) Service Discovery is used. The elements in gray color are configured elements. The corresponding request port (SWC_P1_Req) can be deduced by evaluating the portgroup reference which references the application events (here Ev1). The relation

between application events to the corresponding SD Events (here Ev1 in Eventgroup 1) is available by means of the Ecu Configuration.

[SWS_BswM_00287] Event Subscription Rule Creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each event(-group) of a service instance affected by a request port according to TPS_SWCT_03504 *Software-component acts as a client and subscribe to events and methods* a `BswMRule` shall be created. The shortname of this Rule shall be the shortname of the targeted event(-group) of the service instance prefixed by ESR_.⌋

Note: SomeIP Service Discovery does only support event groups. Therefore, in case SomeIP is used, this rule may cover more than one event.

[SWS_BswM_00288] Service Request Rule Creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each service instance affected by a request port according to TPS_SWCT_03504 a `BswMRule` shall be created. The shortname of this Rule shall be the shortname of the targeted service instance prefixed by SRR_.⌋

Note: Service Request rule collects all request ports targetting one service instance. This way a Client Service will be requested if any event or method is requested.

[SWS_BswM_00289] Client Subscription SenderReceiver Interface

Upstream requirements: SRS_ModeMgm_02008

⌈The SenderReceiverInterface shall look as as defined in [SWS_BSWM_91010]

Where {RuleName} is the name of the corresponding rule and {PortName} is the short-name of the corresponding SWC Port.⌋

Note: The {PortName} is sufficient to match the corresponding request ports.

Note: The same interface is used for both requesting events and requesting the client service.

[SWS_BswM_00290] Client Subscription RPort

Upstream requirements: SRS_ModeMgm_02008

⌈For each request port according to TPS_SWCT_03504 an RPortPrototype according to [SWS_BSWM_91007] shall be created.⌋

[SWS_BswM_00291] Input for Event Subscription Rule

Upstream requirements: SRS_ModeMgm_02008

⌈All request Ports [SWS_BswM_00290] belonging to the same event(-group) shall serve as input for the Event Subscription Rule as in [SWS_BswM_00287].⌋

[SWS_BswM_00292] Input for Service Request Rule

Upstream requirements: SRS_ModeMgm_02008

⌈All request Ports [SWS_BswM_00290] belonging to the same service shall serve as input for the Service Request Rule as in [SWS_BswM_00288].⌋

Note: It can be deduced from EcuC which ports belong to the same requestable entity (e.g. eventgroup and/or client service in case of SomeIP-SD).

[SWS_BswM_00293] Content of Event Subscription rule

Upstream requirements: SRS_ModeMgm_02008

⌈The Event Subscription Rule as of [SWS_BswM_00287] shall be created in the following way:

BswMRuleInitState = FALSE.

The rule shall evaluate to TRUE if at least one associated request port [SWS_BswM_00291] evaluates to REQUEST_SERVICE. If all request ports evaluate to RELEASE_SERVICE the rule shall evaluate to FALSE.⌋

[SWS_BswM_00294] Content of Client Service request rule

Upstream requirements: SRS_ModeMgm_02008

⌈The Service Request Rule as of [SWS_BswM_00288] shall be created in the following way:

BswMRuleInitState = FALSE.

The rule shall evaluate to TRUE if at least one associated request port [SWS_BswM_00291] evaluates to REQUEST_SERVICE. If all request ports evaluate to RELEASE_SERVICE the rule shall evaluate to FALSE.⌋

[SWS_BswM_00295] Client Service Request TRUE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that requests the client service of the corresponding events/methods.

- In case (SomeIP-) Service Discovery is used SD_CLIENT_SERVICE_REQUESTED shall be requested via Sd_ClientServiceSetState() from the SD module.

This Action List shall be the TRUE Action List of the rule as of [SWS_BswM_00294].⌋

[SWS_BswM_00296] Client Service Request FALSE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that releases the request of the client service containing the corresponding events.

- In case (SomeIP-) Service Discovery is used SD_CLIENT_SERVICE_RELEASED shall be requested Sd_ClientServiceSetState() from the SD module.

This Action List shall be the FALSE Action List of the rule as of [SWS_BswM_00294].⌋

[SWS_BswM_00297] Order of ActionList Execution

Upstream requirements: SRS_ModeMgm_02008

⌈The Event Subscription TRUE ActionLists [SWS_BswM_00298] of a Client Service shall always be executed after the Client Service Request TRUE ActionList [SWS_BswM_00295] of that Client Service.⌋

Note to [SWS_BswM_00297]: If an Eventgroup is requested before its Client Service is requested, Service Discovery will not request it. On the other hand releasing a Client Service implicitly leads to releasing all associated Eventgroups.

[SWS_BswM_00298] Event Subscription TRUE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that requests the subscription of the corresponding events.

- In case (SomeIP-) Service Discovery is used SD_CONSUMED_EVENTGROUP_REQUESTED shall be requested via Sd_ConsumedEventGroupSetState() from the SD module.

This Action List shall be the TRUE Action List of the rule as of [SWS_BswM_00293].⌋

[SWS_BswM_00299] Event Subscription FALSE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that requests the unsubscription of the corresponding events.

- In case (SomeIP-) Service Discovery is used SD_CONSUMED_EVENTGROUP_RELEASED shall be requested Sd_ConsumedEventGroupSetState() from the SD module.

This Action List shall be the FALSE Action List of the rule as of [SWS_BswM_00293].⌋

7.8.3 Client Service/Event Subscription state notification

A software-component that acts as a client and is interested in the subscription status for the events and methods exposes an RPortPrototype typed by a ModeSwitchInterface. By this means the software-component can be notified by the BswM whether a subscription is in place.

[SWS_BswM_00300] Mode Switch Interface ClientServiceSubscriptionState

Upstream requirements: SRS_ModeMgm_02008

⌈The Mode Switch Interface shall look as defined in [SWS_BSWM_91016],

Where {RuleName} shall be the of the corresponding rule [SWS_BswM_00301] and {SwcName_PortName} shall be the shortName of the SWC and PortName of the corresponding SWC Port.

The initial mode shall be RELEASE_SERVICE.⌋

Note: The same Interface is used for both notifying event subscriptions and/or indicating availability of the client service, depending on whether events are affected.

[SWS_BswM_00301] Service Availability Notification Rule creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each portgroup being referenced by any RPortPrototype according to TPS_SWCT_03505 *Software-component receives notification on client event and method subscription status changes* a `BswMRule` shall be created. The shortname shall be SANR_{shortName of the PortGroup}.⌋

[SWS_BswM_00302] BswMModeRequestSource BswMSdConsumedEventGroup CurrentState

Upstream requirements: SRS_ModeMgm_02008

⌈For each Consumed Eventgroup affected by a Service Availability Notification Rule [SWS_BswM_00301] a BswMModeRequestSource.BswMSdConsumedEventGroupCurrentState shall be created that serves as input for that rule.⌋

[SWS_BswM_00303] BswMModeRequestSource BswMSdClientServiceCurrentState

Upstream requirements: SRS_ModeMgm_02008

⌈For each Client Service affected by a Service Availability Notification Rule [SWS_BswM_00301] a BswMModeRequestSource.BswMSdClientServiceCurrentState shall be created that serves as input for that rule.⌋

[SWS_BswM_00304] Client Event Subscription notification rule

Upstream requirements: SRS_ModeMgm_02008

⌈For each portgroup being referenced by any RPortPrototype according to TPS_SWCT_03505 one `BswMRule` shall be created in the following way:

BswMRuleInitState = FALSE.

The rule shall evaluate to TRUE if all associated mode request sources [SWS_BswM_00303] evaluate to SD_CLIENT_SERVICE_AVAILABLE and all associated mode request sources [SWS_BswM_00302] evaluate to

SD_CONSUMED_EVENTGROUP_AVAILABLE. Otherwise the rule shall evaluate to FALSE.⌋

Note: Highest wins, only if all Events and all Services associated to a port group are available, the availability of that portgroup is given.

[SWS_BswM_00305] Client Subscription Notification PPort

Upstream requirements: SRS_ModeMgm_02008

⌈For each portgroup being referenced by any RPortPrototype according to TPS_SWCT_03505 a PPortPrototype according to [SWS_BSWM_91015] shall be created.⌋

[SWS_BswM_00306] Client Service availability TRUE action list

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that performs the notification of the corresponding SWCs by setting the current mode of Client Subscription Notification PPort [SWS_BswM_00305] to REQUEST_SERVICE. This Action List shall be the TRUE Action List of the rule as of [SWS_BswM_00304].⌋

[SWS_BswM_00307] Client Service availability FALSE action list

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that performs the notification of the corresponding SWCs by setting the current mode of Client Subscription Notification PPort [SWS_BswM_00305] to RELEASE_SERVICE.

This Action List shall be the FALSE Action List of the rule as of [SWS_BswM_00304].⌋

Note: In a typical scenario 1 event in 1 client service (or only client service in case of a method) will be affected. However, since it is possible to group an arbitrary number of events, methods and even different service (instances) [SWS_BswM_00304] leads to an indication only if all of them are available and subscribed.

7.8.4 Server Service Offer Request

A software-component that implements the server functionality acts as a mode requester and exposes a PPortPrototype typed by a SenderReceiverInterface. By this means, the software-component can send mode requests towards the BswM in order to offer and stop offer that service instance. On the side of the BswM, a RPortPrototype typed by the same SenderReceiver-Interface is used to requests for mode switches.

Unlike the client use cases, the server service offer is only allowed by one port, that means that per instance of a service only one single port is supported.

[SWS_BswM_CONSTR_00007] Unique port for offering a service instance ⌈Per instance of a service only one port according to TPS_SWCT_03502 is allowed.⌋

[SWS_BswM_00308] Sender Receiver Interface ServerServiceRequest

Upstream requirements: SRS_ModeMgm_02008

⌈The SenderReceiverInterface shall look as defined in [SWS_BSWM_91009]

Where {RuleName} is the name of the corresponding rule and {PortName} is the shortname of the corresponding SWC Port of the Server SWC.⌋

Note: The {PortName} is sufficient to match the corresponding request ports.

[SWS_BswM_00309] Service Offer Rule Creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each service instance affected by a request port according to TPS_SWCT_03502 a `BswMRule` shall be created. The shortname of this Rule shall be the shortname of the targeted service prefixed by SOR_.⌋

[SWS_BswM_00310] Service Offer Port Creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each request port according to TPS_SWCT_03502 an RPortPrototype according to [SWS_BSWM_91009] shall be created.⌋

[SWS_BswM_00311] Service Offer Rule Input

Upstream requirements: SRS_ModeMgm_02008

⌈The request Port [SWS_BswM_00301] belonging to the same service instance shall serve as input for the rule as in [SWS_BswM_00309].⌋

Note: Due to [SWS_BswM_CONSTR_00007] only one request port per rule will be created.

Note: It can be deduced from EcuC which port belong to the same service instance.

[SWS_BswM_00312] Service Offer rule

Upstream requirements: SRS_ModeMgm_02008

⌈The Service Offer Rule as of [SWS_BswM_00309] shall be created in the following way:

BswMRuleInitState = FALSE.

The rule shall evaluate to TRUE if all associated request ports [SWS_BswM_00311] evaluate to OFFER_SERVICE. If at least one request port evaluate to STOPOFFER_SERVICE the rule shall evaluate to FALSE.⌋

[SWS_BswM_00313] Service Offer TRUE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that requests the service offer of the corresponding service instance.

- In case (SomeIP-) Service Discovery is used SD_SERVER_SERVICE_AVAILABLE shall be requested via Sd_ServerServiceSetState() from the SD module.

This Action List shall be the TRUE Action List of the rule as of [SWS_BswM_00312].⌋

[SWS_BswM_00314] Service Offer FALSE ActionList

Upstream requirements: SRS_ModeMgm_02008

⌈An Action List shall be created that requests the stop offer of the corresponding service instance.

- In case (SomeIP-) Service Discovery is used SD_SERVER_SERVICE_DOWN shall be requested Sd_ServerServiceSetState() from the SD module.

This Action List shall be the FALSE Action List of the rule as of [SWS_BswM_00312].⌋

7.8.5 Server Service/Event Subscription State Notification

A server instance sending events or field notifications might not need to produce and send those events if there are no subscriptions for that server instance events.

In order to notify the server instance whether there is at least one subscriber interested in the server's events a ModeSwitchInterface is available to receive the subscription notifications.

A software-component that acts as a server and is interested in the subscription status for the service instance events exposes an RPortPrototype typed by a ModeSwitchInterface. By this means, the software-component can be notified by the BswM on subscription changes.

[SWS_BswM_00315] MSIF ServerServiceSubscriptionState

Upstream requirements: SRS_ModeMgm_02008

⌈The Mode Switch Interface shall look as defined in [SWS_BSWM_91013]

Where {RuleName} shall be the of the corresponding rule [SWS_BswM_00316] and {SwcName_PortName} shall be the shortName of the SWC and PortName of the corresponding SWC Port.

The initial mode shall be RELEASE_SERVICE.⌋

[SWS_BswM_00316] Server Event Subscription Notification Rule creation

Upstream requirements: SRS_ModeMgm_02008

⌈For each portgroup being referenced by any RPortPrototype according to TPS_SWCT_03503 *Software-component receives notification on server event subscription status changes* a `BswMRule` shall be created. The shortname shall be SESNR_{shortName of the PortGroup}.⌋

[SWS_BswM_00317] Server Subscription Notification PPort

Upstream requirements: SRS_ModeMgm_02008

⌈For each notification port according to TPS_SWCT_03503 an PPortPrototype according to [SWS_BSWM_91012] shall be created.⌋

[SWS_BswM_00318] BswMModeRequestSource BswMSdEventHandlerCurrentState

Upstream requirements: SRS_ModeMgm_02008

⌈For each Event(-group) affected by a Server Event Subscription Notification Rule [SWS_BswM_00316] a BswMModeRequestSource.BswMSdEventHandlerCurrentState shall be created that serves as input for that rule.⌋

[SWS_BswM_00319] Client Service availability notification rule content

Upstream requirements: SRS_ModeMgm_02008

⌈If at least one BswMModeRequestSource [SWS_BswM_00318] belonging to a Server Event Subscription Notification Rule [SWS_BswM_00316] indicate SD_EVENT_HANDLER_REQUESTED the corresponding ModeSwitchInterface [SWS_BswM_00317] shall have the current mode REQUEST_SERVICE. Otherwise it shall have the current mode RELEASE_SERVICE.⌋

7.8.6 Connecting ports at configuration time

The definition of the actual type of the interfaces and especially the given naming convention facilitate the connection of ports. That means that a tooling can analyze them and automatically connect the ports based on their name. The neccessary attributes for doing so are provided at design time of the SWC-Description via ServiceDependency's.

This is true for ports with the following properties:

- SenderReceiverInterfaces having RoleBasedDataAssignment set to ClientEventSubscription
- ModeSwitchInterfaces having RoleBasedDataAssignment set to ClientEventSubscriptionStatus
- SenderReceiverInterfaces having RoleBasedDataAssignment set to ServerServiceOffer
- ModeSwitchInterfaces having RoleBasedDataAssignment set to ClientEventSubscriptionStatus

8 API specification

8.1 Imported types

[SWS_BswM_00237]

Upstream requirements: SRS_BSW_00301

⌈The BSW Mode Manager shall use only the imported types which are listed in SWS_BswM_00001.⌋

[SWS_BswM_00001] Definition of imported datatypes of module BswM

Upstream requirements: SRS_BSW_00301

⌈

Module	*Header File*	*Imported Type*
CanSM	CanSM.h	CanSM_BswMCurrentStateType
Com	Com.h	Com_IpduGroupIdType
ComM	ComM.h	ComM_PncModeType
	Rte_ComM_Type.h	ComM_ModeType
	Rte_ComM_Type.h	ComM_UserHandleType
Comtype	ComStack_Types.h	NetworkHandleType
	ComStack_Types.h	PNCHandleType
	ComStack_Types.h	PduIdType
Dcm	Dcm.h	Dcm_CommunicationModeType
EcuM	EcuM.h	EcuM_RunStatusType
	EcuM.h	EcuM_StateType
	EcuM.h	EcuM_WakeupSourceType
	EcuM.h	EcuM_WakeupStatusType
	Rte_EcuM_Type.h	EcuM_ShutdownModeType
	Rte_EcuM_Type.h	EcuM_ShutdownTargetType
	Rte_EcuM_Type.h	EcuM_UserType
Eth	Eth_GeneralTypes.h	Eth_ModeType
EthIf	EthIf.h	EthIf_SwitchPortGroupIdxType
EthSM	EthSM.h	EthSM_NetworkModeStateType
EthTrcv	Eth_GeneralTypes.h	EthTrcv_LinkStateType
FrSM	FrSM.h	FrSM_BswM_StateType
IdsM	IdsM.h	IdsM_Filters_BlockStateType
J1939Dcm	J1939Dcm.h	J1939Dcm_StateType
J1939Rm	J1939Rm.h	J1939Rm_StateType
LinIf	LinIf.h	LinIf_SchHandleType
	LinIf.h	LinTp_Mode
LinSM	LinSM.h	LinSM_ModeType
Nm	NmStack_types.h	Nm_StateType
NvM	NvM.h	NvM_MultiBlockRequestType

▽

△

Module	Header File	Imported Type
	Rte_NvM_Type.h	NvM_BlockIdType
	Rte_NvM_Type.h	NvM_RequestResultType
PduR	PduR.h	PduR_RoutingPathGroupIdType
Sd	Sd.h	Sd_ClientServiceCurrentStateType
	Sd.h	Sd_ClientServiceSetStateType
	Sd.h	Sd_ConsumedEventGroupCurrentStateType
	Sd.h	Sd_ConsumedEventGroupSetStateType
	Sd.h	Sd_EventHandlerCurrentStateType
	Sd.h	Sd_ServerServiceSetStateType
	Sd.h	Sd_ServiceGroupIdType
SoAd	SoAd.h	SoAd_SoConIdType
	SoAd.h	SoAd_SoConModeType
Std	Std_Types.h	Std_ReturnType

⌋

8.2 Type definitions

[SWS_BswM_00041]

Upstream requirements: SRS_BSW_00305

⌈The following Data Types shall be used for the functions defined in this specification.⌋

8.2.1 BswM_ConfigType

[SWS_BswM_00213] Definition of datatype BswM_ConfigType

Upstream requirements: SRS_BSW_00404, SRS_BSW_00441

⌈

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

⌋

[SWS_BswM_00042]

Upstream requirements: SRS_BSW_00404

⌈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

[SWS_BswM_00214] Definition of datatype BswM_ModeType

Upstream requirements: SRS_ModeMgm_02003, SRS_BSW_00441

⌈

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

⌋

8.2.3 BswM_UserType

[SWS_BswM_00216] Definition of datatype BswM_UserType

Upstream requirements: SRS_ModeMgm_02001, SRS_BSW_00441

⌈

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

⌋

8.3 Function definitions

This chapter contains all APIs provided by the module.

8.3.1 BswM_CanSM_CurrentState

[SWS_BswM_00049] Definition of API function BswM_CanSM_CurrentState

Upstream requirements: SRS_ModeMgm_02001

⌈

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.	
Available via	BswM_CanSM.h	

⌋

8.3.2 BswM_ComM_CurrentMode

[SWS_BswM_00047] Definition of API function BswM_ComM_CurrentMode

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.3 BswM_ComM_CurrentPNCMode

[SWS_BswM_00148] Definition of API function BswM_ComM_CurrentPNCMode

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.4 BswM_ComM_InitiateReset

[SWS_BswM_00217] Definition of API function BswM_ComM_InitiateReset

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.5 BswM_Dcm_ApplicationUpdated

[SWS_BswM_00158] Definition of API function BswM_Dcm_ApplicationUpdated

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.6 BswM_Dcm_CommunicationMode_CurrentState

[SWS_BswM_00048] Definition of API function BswM_Dcm_CommunicationMode_CurrentState

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_Dcm_CommunicationMode_CurrentState	
Syntax	`void BswM_Dcm_CommunicationMode_CurrentState ( NetworkHandleType Network, Dcm_CommunicationModeType RequestedMode )`	
Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Network	The communication channel that the diagnostic mode corresponds to.
	RequestedMode	The requested diagnostic communication mode.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Function called by DCM to inform the BswM about the current state of the communication mode.	
Available via	BswM_Dcm.h	

⌋

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

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

8.3.7 BswM_Deinit

[SWS_BswM_00119] Definition of API function BswM_Deinit

Upstream requirements: SRS_BSW_00336

⌈

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

⌋

[SWS_BswM_00120]

Upstream requirements: SRS_BSW_00336

⌈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.⌋

8.3.8 BswM_EcuM_CurrentState

[SWS_BswM_91003] Definition of API function BswM_EcuM_CurrentState

Upstream requirements: SRS_ModeMgm_02001, SRS_ModeMgm_02005, SRS_ModeMgm_02006

⌈

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

⌋

8.3.9 BswM_EcuM_CurrentWakeup

[SWS_BswM_00131] Definition of API function BswM_EcuM_CurrentWakeup

Upstream requirements: SRS_ModeMgm_02001, SRS_ModeMgm_02005, SRS_ModeMgm_02006

⌈

Service Name	BswM_EcuM_CurrentWakeup	
Syntax	`void BswM_EcuM_CurrentWakeup ( EcuM_WakeupSourceType source, EcuM_WakeupStatusType state )`	
Service ID [hex]	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	source	Wakeup source(s) that changed state.
	state	The new state of the wakeup source(s)
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Function called by EcuM to indicate the current state of a wakeup source.	
Available via	BswM_EcuM.h	

⌋

8.3.10 BswM_EcuM_RequestedState

[SWS_BswM_91004] Definition of API function BswM_EcuM_RequestedState

Upstream requirements: SRS_ModeMgm_02001, SRS_ModeMgm_02005, SRS_ModeMgm_02006

⌈

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

⌋

8.3.11 BswM_EthIf_PortGroupLinkStateChg

[SWS_BswM_91001] Definition of API function BswM_EthIf_PortGroupLinkStateChg

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_EthIf_PortGroupLinkStateChg	
Syntax	`void BswM_EthIf_PortGroupLinkStateChg ( EthIf_SwitchPortGroupIdxType PortGroupIdx, EthTrcv_LinkStateType PortGroupState )`	
Service ID [hex]	0x26	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	PortGroupIdx	The port group index in the context of the Ethernet Interface
	PortGroupState	The state of the port group. State is derived from the physical link of the Ethernet Transceiver: ETHTRCV_LINK_STATE_DOWN == Port group has link down. ETHTRCV_LINK_STATE_ACTIVE == Port group has link up.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	

▽

△

Description	Function called by EthIf to indicate the link state change of a certain Ethernet switch port group.
Available via	BswM_EthIf.h

⌋

8.3.12 BswM_EthSM_CurrentState

[SWS_BswM_00050] Definition of API function BswM_EthSM_CurrentState

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.13 BswM_FrSM_CurrentState

[SWS_BswM_00051] Definition of API function BswM_FrSM_CurrentState

Upstream requirements: SRS_ModeMgm_02001

⌈

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

▽

△

	CurrentState	The corrent state of the FlexRay cluster.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Function called by FrSM to indicate its current state.	
Available via	BswM_FrSM.h	

⌋

8.3.14 BswM_Init

[SWS_BswM_00002] Definition of API function BswM_Init

Upstream requirements: SRS_BSW_00344, SRS_BSW_00404, SRS_BSW_00405, SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_00414

⌈

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

⌋

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

[SWS_BswM_00043]

Upstream requirements: SRS_BSW_00101

⌈This routine initializes the BSW Mode Manager. After execution of this routine the BSW Mode Manager is ready to arbitrate incoming mode requests.⌋

[SWS_BswM_00044]

Upstream requirements: SRS_BSW_00101

⌈This routine shall initialize all module global variables of the BSW Mode Manager.⌋

[SWS_BswM_00118]

Upstream requirements: SRS_BSW_00467

⌈BswM_Init shall only require the OS and the SchM to be initialized before it can be called.⌋

8.3.15 BswM_J1939DcmBroadcastStatus

[SWS_BswM_00165] Definition of API function BswM_J1939DcmBroadcastStatus

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_J1939DcmBroadcastStatus	
Syntax	`void BswM_J1939DcmBroadcastStatus ( uint16 NetworkMask )`	
Service ID [hex]	0x1b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkMask	Mask containing one bit for each available network. The bit position within this mask corresponds to the ComMChannel.ComMChannelId for the communication channel (so ComMChannelID 0 is represented by bit 0). The meaning for each bit is: 1: Network enabled, 0: Network disabled. Note: only the first 16 communication channel IDs can be supported by this API.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This API tells the BswM the desired communication status of the available networks. The status will typically be activated via COM I-PDU group switches.	
Available via	BswM_J1939Dcm.h	

⌋

[SWS_BswM_00249]

Upstream requirements: SRS_ModeMgm_02001

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

8.3.16 BswM_J1939Nm_StateChangeNotification

[SWS_BswM_00194] Definition of API function BswM_J1939Nm_StateChange Notification

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.17 BswM_LinSM_CurrentSchedule

[SWS_BswM_00058] Definition of API function BswM_LinSM_CurrentSchedule

Upstream requirements: SRS_ModeMgm_02001

⌈

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

▽

△

Available via	BswM_LinSM.h

⌋

8.3.18 BswM_LinSM_CurrentState

[SWS_BswM_00052] Definition of API function BswM_LinSM_CurrentState

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.19 BswM_LinTp_RequestMode

[SWS_BswM_00156] Definition of API function BswM_LinTp_RequestMode

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_LinTp_RequestMode	
Syntax	`void BswM_LinTp_RequestMode ( NetworkHandleType Network, LinTp_Mode LinTpRequestedMode )`	
Service ID [hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Network	The LIN channel that the LinTp mode request relates to.
	LinTpRequestedMode	The requested LIN TP mode.

▽

△

Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Function called by LinTP to request a mode for the corresponding LIN channel. The LinTp_Mode correlates to the LIN schedule table that should be used.
Available via	BswM_LinTp.h

⌋

8.3.20 BswM_Mka_ControlledPortStatus

[SWS_BswM_91019] Definition of API function BswM_Mka_ControlledPortStatus

⌈

Service Name	BswM_Mka_ControlledPortStatus	
Syntax	`void BswM_Mka_ControlledPortStatus ( NetworkHandleType Network, uint8 SecYIdx, boolean ControlledPortEnabled )`	
Service ID [hex]	0x2b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Network	The network channel that the indicated state corresponds to.
	SecYIdx	SecY for which the status is reported
	ControlledPortEnabled	TRUE - The Controlled Port for this SecY is enabled FALSE - The Controlled Port for this SecY is disabled
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Function called by MKA to indicate the current state of a Controlled Port for a SecY.	
Available via	BswM_Mka.h	

⌋

8.3.21 BswM_Nm_CarWakeUpIndication

[SWS_BswM_00235] Definition of API function BswM_Nm_CarWakeUpIndication

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_Nm_CarWakeUpIndication
Syntax	`void BswM_Nm_CarWakeUpIndication ( NetworkHandleType Network )`

▽

△

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

⌋

8.3.22 BswM_Nm_StateChangeNotification

[SWS_BswM_91002] Definition of API function BswM_Nm_StateChangeNotification

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.23 BswM_NvM_CurrentBlockMode

[SWS_BswM_00104] Definition of API function BswM_NvM_CurrentBlockMode

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.24 BswM_NvM_CurrentJobMode

[SWS_BswM_00152] Definition of API function BswM_NvM_CurrentJobMode

Upstream requirements: SRS_ModeMgm_02001

⌈

Service Name	BswM_NvM_CurrentJobMode	
Syntax	`void BswM_NvM_CurrentJobMode ( NvM_MultiBlockRequestType MultiBlockRequest, NvM_RequestResultType CurrentJobMode )`	
Service ID [hex]	0x17	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	MultiBlockRequest	Indicates which multi block service this callback refers to.
	CurrentJobMode	Current state of the multi block job indicated by parameter Multi BlockRequest
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Function called by NvM to inform the BswM about the current state of a multi block job.	
Available via	BswM_NvM.h	

⌋

8.3.25 BswM_RequestMode

[SWS_BswM_00046] Definition of API function BswM_RequestMode

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.3.26 BswM_Sd_ClientServiceCurrentState

[SWS_BswM_00204] Definition of API function BswM_Sd_ClientServiceCurrentState

Upstream requirements: SRS_ModeMgm_02008

⌈

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

▽

△

Available via	BswM_Sd.h

⌋

8.3.27 BswM_Sd_ConsumedEventGroupCurrentState

[SWS_BswM_00207] Definition of API function BswM_Sd_ConsumedEvent GroupCurrentState

Upstream requirements: SRS_ModeMgm_02008

⌈

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

⌋

8.3.28 BswM_Sd_EventHandlerCurrentState

[SWS_BswM_00210] Definition of API function BswM_Sd_EventHandlerCurrent State

Upstream requirements: SRS_ModeMgm_02008

⌈

Service Name	BswM_Sd_EventHandlerCurrentState
Syntax	`void BswM_Sd_EventHandlerCurrentState ( uint16 SdEventHandlerHandleId, Sd_EventHandlerCurrentStateType EventHandlerStatus )`
Service ID [hex]	0x20

▽

△

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

⌋

8.3.29 BswM_SoAd_SoConModeChg

[SWS_BswM_91005] Definition of callback function BswM_SoAd_SoConMode Chg

Upstream requirements: SRS_ModeMgm_02008

⌈

Service Name	BswM_SoAd_SoConModeChg	
Syntax	`void BswM_SoAd_SoConModeChg ( SoAd_SoConIdType SoConId, SoAd_SoConModeType Mode )`	
Service ID [hex]	0x2a	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	SoConId	Socket connection index specifying the socket connection with the mode change.
	Mode	New socket connection mode.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Notification about a SoAd socket connection mode change, e.g. socket connection gets online.	
Available via	BswM_SoAd.h	

⌋

8.4 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.

8.4.1 BswM_MainFunction

[SWS_BswM_00053] Definition of scheduled function BswM_MainFunction

Upstream requirements: SRS_BSW_00373

⌈

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

⌋

[SWS_BswM_00075]

Upstream requirements: SRS_ModeMgm_02000, SRS_ModeMgm_02002

⌈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.⌋

8.5 Expected Interfaces

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

8.5.1 Mandatory Interfaces

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

[SWS_BswM_00007] Definition of mandatory interfaces required by module Bsw M

Upstream requirements: SRS_BSW_00384

⌈

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

⌋

8.5.2 Optional Interfaces

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

[SWS_BswM_00008] Definition of optional interfaces requested by module BswM

Upstream requirements: SRS_BSW_00384

⌈

API Function	*Header File*	*Description*
Com_DisableReceptionDM	Com.h	Disables the reception deadline monitoring for the I-PDUs within the given I-PDU group.
Com_EnableReceptionDM	Com.h	Enables the reception deadline monitoring for the I-PDUs within the given I-PDU group.
Com_IpduGroupStart	Com.h	Starts a preconfigured I-PDU group. For example, cyclic I-PDUs will be sent out cyclically after the call of Com_IpduGroupStart(). If Initialize is true all I-PDUs of the I-PDU group shall be (re-)initialized before the I-PDU group is started. That is they shall behave like after a start-up of COM, for example the old_value of the filter objects and shadow buffers of signal groups have to be (re-)initialized.
Com_IpduGroupStop	Com.h	Stops a preconfigured I-PDU group. For example, cyclic I-PDUs will be stopped after the call of Com_IpduGroupStop().
Com_SwitchIpduTxMode	Com.h	The service Com_SwitchIpduTxMode sets the transmission mode of the I-PDU referenced by Pdu Id to Mode. In case the transmission mode changes, the new mode shall immediately be effective (see [SWS_Com_00239]). In case the requested transmission mode was already active for this I-PDU, the call will have no effect.
Com_TriggerIPDUSend	Com.h	By a call to Com_TriggerIPDUSend the I-PDU with the given ID is triggered for transmission.
ComM_CommunicationAllowed	ComM_BswM.h	EcuM or BswM shall indicate to ComM when communication is allowed. If EcuM/Flex is used: BswM
ComM_LimitChannelToNoComMode	ComM.h	Changes the inhibition status for the channel for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_LimitECUToNoComMode, same functionality but for all channels)
ComM_LimitECUToNoComMode	ComM.h	Changes the inhibition status for the ECU (=all channels) for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_LimitChannelToNoComMode, same functionality but for a specific channels)

▽

△

API Function	*Header File*	*Description*
ComM_RequestComMode	ComM.h	Requesting of a Communication Mode by a user. Note: The following modes are invalid user requests, since they are used as internal modes: • COMM_SILENT_COMMUNICATION (this mode is used for synchronization at shutdown) • COMM_FULL_COMMUNICATION_WITH_ WAKEUP_REQUEST (this mode is used internally within the ComM channel statemachine to trigger the lower layers to request a wakeup on the network if the used hardware support such a feature. (e.g. Ethernet hardware which is compatible with OA TC10). The following modes are valid user requests: • COMM_NO_COMMUNICATION • COMM_FULL_COMMUNICATION. The communication request could also be released due to a ComM communication inhibition
Det_ReportError	Det.h	Service to report development errors.
EcuM_AL_DriverInitBswM_<x>	EcuM.h	This callback shall provide BSW module initializations to be called by the BSW Mode Manager.
EcuM_GoDownHaltPoll	EcuM.h	Instructs the ECU State Manager module to go into a sleep mode, Reset or OFF depending on the previously selected shutdown target.
EcuM_SelectShutdownTarget	EcuM.h	EcuM_SelectShutdownTarget selects the shutdown target. EcuM_SelectShutdownTarget is part of the ECU Manager Module port interface.
EcuM_SetState	EcuM.h	Function called by BswM to notify about State Switch.
EthIf_StartAllPorts	EthIf.h	Request to set all configured and affected EthSwt Ports to ETH_MODE_ACTIVE
EthIf_SwitchPortGroupRequestMode	EthIf.h	Request a mode for the EthIfSwtPortGroup. The call shall be forwarded to EthSwt by calling EthSwt_Set SwitchPortMode for all EthSwtPorts referenced by the port group.
FrSM_AllSlots	FrSM.h	This API function can be used to leave the KeySlot OnlyMode.
Fw_SetFirewallState (draft)	Fw.h	This function is invoked by the BswM to indicate ECU state changes. **Tags:** atp.Status=draft
IdsM_BswM_StateChanged	IdsM_BswM.h	This callback function is invoked by the BswM to indicate ECU state changes.
J1939Dcm_SetState	J1939Dcm.h	Changes the communication state of J1939Dcm to offline or online.
J1939Rm_SetState	J1939Rm.h	Changes the communication state of J1939Rm to offline (only Request for AC supported) or online.
LinSM_ScheduleRequest	LinSM.h	The upper layer requests a schedule table to be changed on one LIN network.

▽

△

API Function	*Header File*	*Description*
Nm_DisableCommunication	Nm.h	Disables the NM PDU transmission ability. For that purpose <Bus>Nm_DisableCommunication shall be called in case NmBusType is not set to NM_ BUSNM_LOCALNM (e.g. CanNm_Disable Communication function is called if channel is configured as CAN).
Nm_EnableCommunication	Nm.h	Enables the NM PDU transmission ability. For that purpose <Bus>Nm_EnableCommunication shall be called in case NmBusType is not set to NM_ BUSNM_LOCALNM. (e.g. CanNm_Enable Communication function is called if channel is configured as CAN).
PduR_DisableRouting	PduR.h	Disables a routing path group.
PduR_EnableRouting	PduR.h	Enables a routing path group.
Rte_Start_<EcucPartition>	Rte.h	Rte_Start is intended to allocate and initialize system resources and communication resources used by the RTE in the given partition.
Rte_Stop_<EcucPartition>	Rte.h	Rte_Stop is used to finalize the RTE on the partition it is called. This service releases all system and communication resources allocated by the RTE on that partition.
Sd_ClientServiceSetState	Sd.h	This API function is used by the BswM to set the Client Service Instance state.
Sd_ConsumedEventGroupSetState	Sd.h	This API function is used by the BswM to set the requested state of the EventGroupStatus.
Sd_ServerServiceSetState	Sd.h	This API function is used by the BswM to set the Server Service Instance state.
Sd_ServiceGroupStart	Sd.h	Starts a preconfigured SdServiceGroup. For example, OfferService entries will be sent out after the call of Sd_ServiceGroupStart() for all Server Servives of a SdServiceGroup, which are not requested yet.
Sd_ServiceGroupStop	Sd.h	Stops a preconfigured SdServiceGroup. For example, StopOfferService entries will be sent out after the call of Sd_ServiceGroupStop() for all ServerServices of a SdServiceGroup, which are not requested by another SdServiceGroup.

⌋

8.6 Service Interfaces

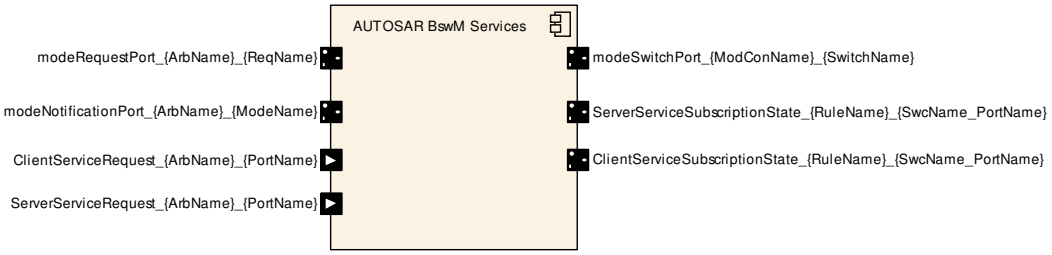


Figure 8.1: Ports provided/required by the BswM service SW-C

8.6.1 Scope of this Chapter

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

8.6.2 Ports

8.6.2.1 BswM_modeNotificationPort

[SWS_BswM_00200] Definition of Port modeNotificationPort_{ArbName}_{Mode Name} required by module BswM

Upstream requirements: SRS_ModeMgm_02001

⌈

Name	modeNotificationPort_{ArbName}_{ModeName}
Kind	RequiredPort
Interface-Ref	{ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMModeRequestSource/BswMSwcModeNotification.BswMSwcModeNotificationModeDeclarationGroupPrototype Ref)}.parent
Description	–
Variation	ArbName = {ecuc(BswM/BswMConfig/BswMArbitration.SHORT-NAME)} ModeName = {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMMode RequestSource/BswMSwcModeNotification.SHORT-NAME)}

⌋

8.6.2.2 BswM_modeRequestPort

[SWS_BswM_00201] Definition of Port modeRequestPort_{ArbName}_{Req Name} required by module BswM

Upstream requirements: SRS_ModeMgm_02001

⌈

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

⌋

8.6.2.3 BswM_modeSwitchPort

[SWS_BswM_00202] Definition of Port modeSwitchPort_{ModConName}_{SwitchName} provided by module BswM

Upstream requirements: SRS_ModeMgm_02004

⌈

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

⌋

8.6.2.4 S/R Port BswM_ClientServiceRequest

[SWS_BSWM_91007] Definition of Port ClientServiceRequest_{ArbName}_{PortName} required by module BswM

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ClientServiceRequest_{ArbName}_{PortName}		
Kind	RequiredPort	***Interface***	ClientServiceRequest_{ArbName}_{PortName}
Description	RPort as stated in [SWS_BswM_00290]		
Variation	–		

⌋

8.6.2.5 S/R Port BswM_ServerServiceRequest

[SWS_BSWM_91011] Definition of Port ServerServiceRequest_{ArbName}_{PortName} required by module BswM

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ServerServiceRequest_{ArbName}_{PortName}		
Kind	RequiredPort	***Interface***	ServerServiceRequest_{ArbName}_{PortName}
Description	RPort as stated in [SWS_BswM_00310]		
Variation	–		

⌋

8.6.2.6 Mode Switch Port BswM_ClientServiceSubscriptionState

[SWS_BSWM_91015] Definition of Port ClientServiceSubscriptionState_{RuleName}_{SwcName_PortName} provided by module BswM

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ClientServiceSubscriptionState_{RuleName}_{SwcName_PortName}		
Kind	ProvidedPort	***Interface***	ClientServiceSubscriptionState
Description	Mode Switch Port indicating the subscription state to the client service offered by the given SWC and Port		
Variation	{RuleName} {SwcName_PortName}		

⌋

8.6.2.7 Mode Switch Port BswM_ServerServiceSubscriptionState

[SWS_BSWM_91012] Definition of Port ServerServiceSubscriptionState_{RuleName}_{SwcName_PortName} provided by module BswM

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ServerServiceSubscriptionState_{RuleName}_{SwcName_PortName}		
Kind	ProvidedPort	***Interface***	ServerServiceSubscriptionState
Description	Mode Switch Port indicating the subscription state to the server service offered by the given SWC and Port		
Variation	{RuleName} {SwcName_PortName}		

⌋

8.6.3 Mode Declaration Groups

8.6.3.1 ModeDeclarationGroup MDG_ClientServiceSubscriptionState

[SWS_BSWM_91018] Definition of ModeDeclarationGroup MDG_ClientService SubscriptionState

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	MDG_ClientServiceSubscriptionState
Kind	ModeDeclarationGroup
Category	ALPHABETIC_ORDER
Initial mode	RELEASE_SERVICE
On transition value	–

▽

△

Modes	RELEASE_SERVICE	–
	REQUEST_SERVICE	–
Description	–	

⌋

8.6.3.2 ModeDeclarationGroup MDG_ServerServiceSubscriptionState

[SWS_BSWM_91014] Definition of ModeDeclarationGroup MDG_ServerService SubscriptionState

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	MDG_ServerServiceSubscriptionState	
Kind	ModeDeclarationGroup	
Category	ALPHABETIC_ORDER	
Initial mode	RELEASE_SERVICE	
On transition value	–	
Modes	RELEASE_SERVICE	–
	REQUEST_SERVICE	–
Description	–	

⌋

8.6.4 ModeSwitchInterfaces

8.6.4.1 ModeSwitchInterface ClientServiceSubscriptionState

[SWS_BSWM_91016] Definition of ModeSwitchInterface ClientServiceSubscrip-tionState

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ClientServiceSubscriptionState	
Comment	mode MDG_ClientServiceSubscriptionState appMode	
IsService	true	
Variation	–	
ModeGroup	MDG_ClientServiceSubscriptionState	MDG_ClientServiceSubscriptionState

⌋

8.6.4.2 ModeSwitchInterface ServerServiceSubscriptionState

[SWS_BSWM_91013] Definition of ModeSwitchInterface ServerServiceSubscriptionState

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ServerServiceSubscriptionState	
Comment	mode MDG_ServerServiceSubscriptionState appMode	
IsService	true	
Variation	–	
ModeGroup	MDG_ServerServiceSubscriptionState	MDG_ServerServiceSubscriptionState

⌋

8.6.5 S/R Interfaces

8.6.5.1 S/R interface ClientServiceRequest

[SWS_BSWM_91010] Definition of SenderReceiverInterface ClientServiceRequest_{ArbName}_{PortName}

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ClientServiceRequest_{ArbName}_{PortName}	
Comment	Interface for requesting a Service / Subscription to an Event	
IsService	true	
Variation	{ArbName}_{PortName}	
Data Elements	ClientServiceSetStateRequest	
	Type	BswM_ClientModeRequestType
	Variation	–

⌋

8.6.5.2 S/R interface ServerServiceRequest

[SWS_BSWM_91009] Definition of SenderReceiverInterface ServerServiceRequest_{ArbName}_{PortName}

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	ServerServiceRequest_{ArbName}_{PortName}	
Comment	Interface for offering a service	
IsService	true	
Variation	{ArbName}_{PortName}	
Data Elements	ServerServiceSetStateRequest	
	Type	BswM_OfferModeType
	Variation	–

⌋

8.6.6 Types

8.6.6.1 BswM_ClientModeRequestType

[SWS_BSWM_91006] Definition of ImplementationDataType BswM_ClientModeRequestType

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	BswM_ClientModeRequestType		
Kind	Type		
Derived from	uint8		
Range	RELEASE_SERVICE	0x00	Service/Subscription released
	REQUEST_SERVICE	0x01	Service/Subscription requested/ active
Description	Enumeration for requesting subscription as well as querying the subscription state		
Variation	–		
Available via	BswM.h		

⌋

8.6.6.2 BswM_OfferModeType

[SWS_BSWM_91008] Definition of ImplementationDataType BswM_OfferMode Type

Upstream requirements: SRS_ModeMgm_02008

⌈

Name	BswM_OfferModeType		
Kind	Type		
Derived from	uint8		
Range	STOPOFFER_SERVICE	0x00	Service shall not be offered
	OFFER_SERVICE	0x01	Service shall be offered
Description	Enumeration for offering a service		
Variation	–		
Available via	BswM.h		

⌋

8.7 API to Request Port Mappings

[SWS_BswM_00283] API to Request Port Mappings: This table describes which request port shall be triggered when an API is called. In addition, it describes which configuration parameter(s) correspond to which API parameter(s)

Upstream requirements: SRS_ModeMgm_02001

⌈

API	Request Port	API / Config-parameter pairs
BswM_CanSM_CurrentState	BswMCanSMIndication	Network / BswMCanSMChannelRef
BswM_ComM_CurrentMode	BswMComMIndication	Network / BswMComMChannelRef
BswM_ComM_CurrentPNCMode	BswMComMPncRequest	PNC / BswMComMPncRef
BswM_ComM_InitiateReset	BswMComMInitiateReset	-
BswM_Dcm_ApplicationUpdated	BswMDcmApplicationUpdatedIndication	-
BswM_Dcm_CommunicationMode_CurrentState	BswMDcmComModeRequest	Network / BswMDcmComMChannelRef
BswM_EcuM_CurrentState	BswMEcuMIndication	-
BswM_EcuM_CurrentWakeup	BswMEcuMWakeupSource	source / BswMEcuMWakeupSrcRef
BswM_EcuM_RequestedState	BswMEcuMRUNRequestIndication	State / BswMEcuMRUNRequestProtocolPort
BswM_EthIf_PortGroupLinkStateChg	BswMEthIfPortGroupLinkStateChg	PortGroupIdx / BswMEthIfSwitchPortGroupRef
BswM_EthSM_CurrentState	BswMEthSMIndication	Network / BswMEthSMChannelRef
BswM_FrSM_CurrentState	BswMFrSMIndication	Network / BswMFrSMChannelRef

API	Request Port	API / Config-parameter pairs
BswM_J1939DcmBroadcast Status	BswMJ1939DcmBroadcast Status	NetworkMask / BswMJ1939 DcmChannelRef
BswM_J1939Nm_StateChange Notification	BswMJ1939NmIndication	Network / BswMJ1939NmChannelRef, Node / BswMJ1939Nm NodeRef
BswM_LinSM_CurrentSchedule	BswMLinScheduleIndication	Network / BswMLinSMChannel Ref
BswM_LinSM_CurrentState	BswMLinSMIndication	Network / BswMLinSMChannel Ref
BswM_LinTp_RequestMode	BswMLinTpModeRequest	Network / BswMLinTpChannel Ref
BswM_Mka_ControlledPortStatus	BswMMkaControlledPortStatus Indication	Network / BswMMkaChannel Ref
BswM_Nm_CarWakeUpIndication	BswMNmCarWakeUpIndication	-
BswM_Nm_StateChangeNotification	BswMNmStateChangeNotification	Network / BswMNmChannelRef
BswM_NvM_CurrentBlockMode	BswMNvMRequest	Block / BswMNvMBlockRef
BswM_NvM_CurrentJobMode	BswMNvMJobModeIndication	MultiBlockRequest / BswMNvm Service
BswM_RequestMode	BswMGenericRequest	requesting_user / BswMMode RequesterId
BswM_Sd_ClientServiceCurrentState	BswMSdClientServiceCurrent State	SdClientServiceHandleId / BswMSdClientMethodsRef
BswM_Sd_ConsumedEvent GroupCurrentState	BswMSdConsumedEventGroup CurrentState	SdConsumedEventGroupHandleId / BswMSdConsumedEvent GroupRef
BswM_Sd_EventHandlerCurrentState	BswMSdEventHandlerCurrent State	SdEventHandlerHandleId / BswMSdEventHandlerRef
BswM_SoAd_SoConModeChg	BswMSoAdSoConModeChg	SoConId / BswMSoAdSocket-ConnectionRef

9 Sequence diagrams

9.1 Deferred operation of BswM

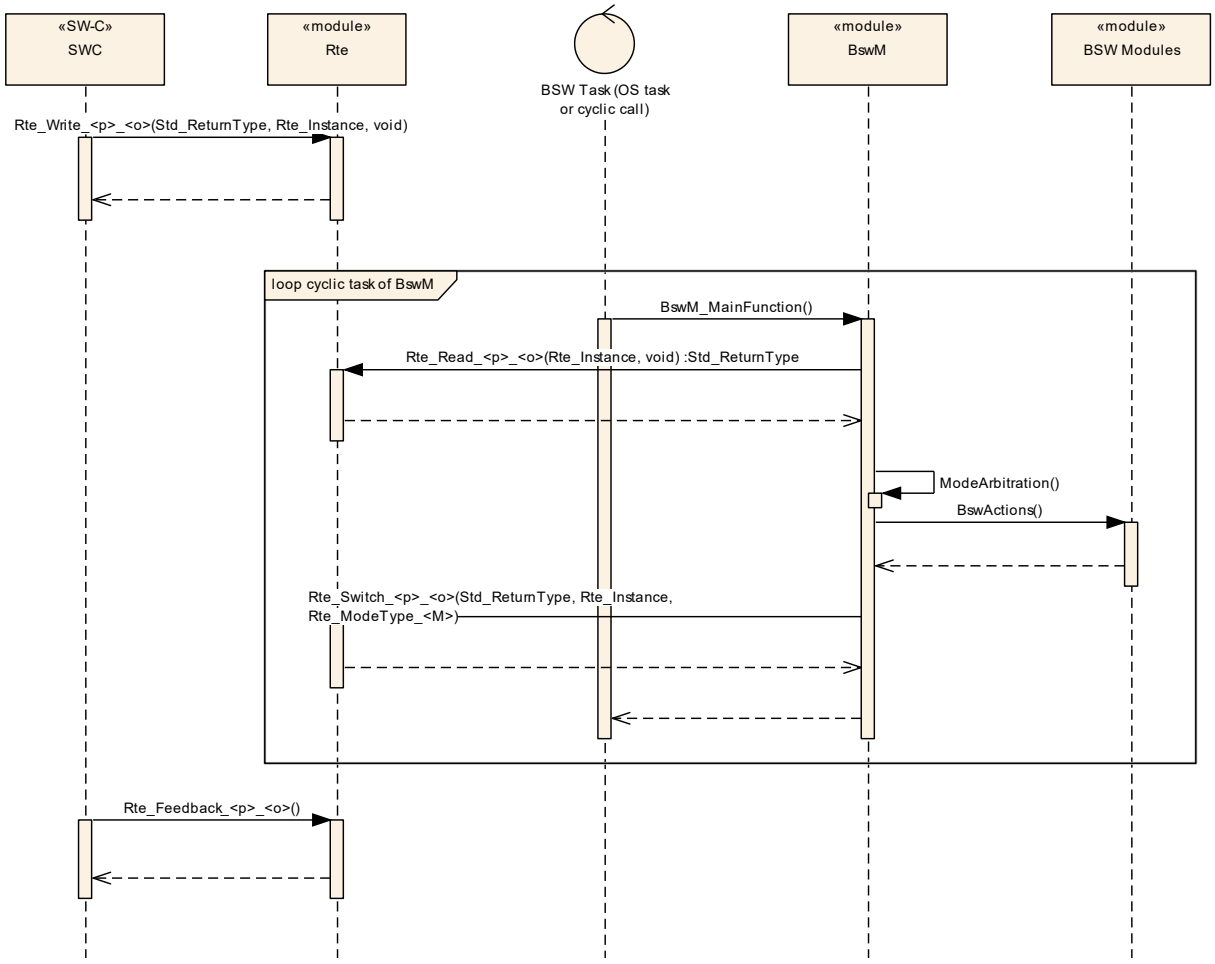


Figure 9.1: Deferred operation of BswM

9.2 Immediate operation of BswM

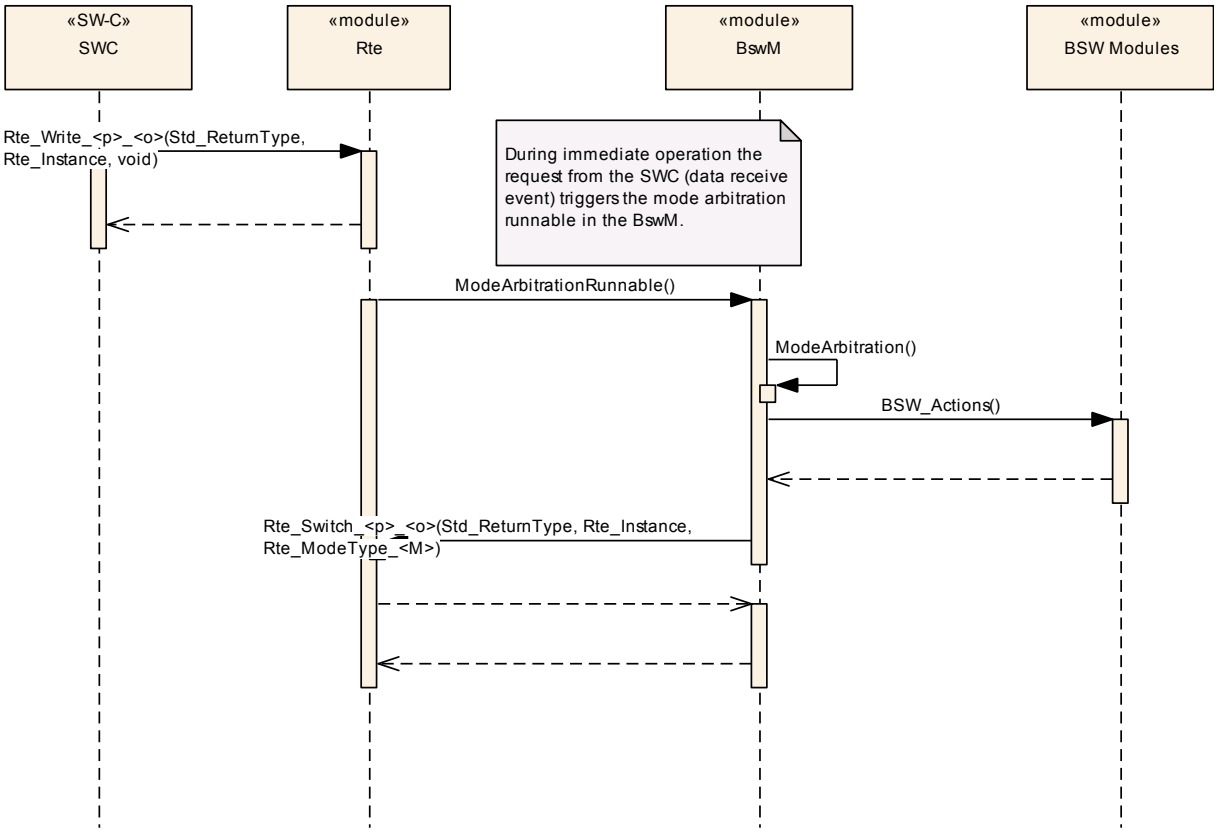


Figure 9.2: Immediate operation of BswM

10 Configuration specification

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

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

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

10.1 How to read this chapter

For details refer to [1] Chapter 10.1 *"Introduction to configuration specification"*.

10.2 Containers and configuration parameters

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

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

10.2.1 BswM

[ECUC_BswM_01063] Definition of EcucModuleDef BswM ⌈

Module Name	BswM
Description	Configuration of the BswM (Basic SW Mode Manager) module.
Post-Build Selectable Support	true
Post-Build Loadable Support	true

Included Containers		
Container Name	**Multiplicity**	**Description**
BswMConfig	1..*	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition.
BswMGeneral	1	General configuration parameters of the Basic SW Mode Manager.

⌋

10.2.2 BswMConfig

[ECUC_BswM_00895] Definition of EcucParamConfContainerDef BswMConfig ⌈

Container Name	BswMConfig
Parent Container	BswM
Upstream mapping defined	–
Description	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition.
Multiplicity	1..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMPartitionRef	0..1	[ECUC_BswM_00984]

Included Containers		
Container Name	**Multiplicity**	**Description**
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.

⌋

[ECUC_BswM_00984] Definition of EcucReferenceDef BswMPartitionRef ⌈

Parameter Name	BswMPartitionRef
Parent Container	BswMConfig
Upstream mapping defined	–
Description	This references the partition the BswM shall run inside.
Multiplicity	0..1
Type	Reference to EcucPartition
Pre-Compile only	true

⌋

10.2.3 BswMArbitration

[ECUC_BswM_00801] Definition of EcucParamConfContainerDef BswMArbitration ⌈

Container Name	BswMArbitration
Parent Container	BswMConfig
Upstream mapping defined	–
Description	This container includes all configuration sub-containers and parameters related to the mode arbitration functionality of the BswM.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
BswMEventRequestPort	0..*	Each instance of this container defines an event which can be sent to the BswM. Basic Software Modules may send these events to the BswM by calling the corresponding BswM C-API (for example: BswM_ComM_InitiateReset()).
BswMLogicalExpression	0..*	This container describes the logical expressions that can be used for the mode arbitration. The logical expressions are built of a set of arguments and a logical operator. Each argument can either be a mode condition or a sub-expression to allow definition of more complex logical expressions. There may be an unlimited number of arguments in each logical expression. Note that the order of evaluation of the expressions is not defined.
BswMModeCondition	0..*	This container describes the BswM mode conditions that can be used either by itself to form a rule or as a part of a logical expression.
BswMModeRequestPort	0..*	Each instance of this container defines a mode request interface that is used to requests or indicate modes from/to the BswM. These interfaces are implemented as ports or as ordinary C-functions based upon if the request is made by an SW-C or a BSW module. There are different types of mode requests: 1. Mode requests from the SW-C:s 2. Mode Requests from other BSW modules such as the DCM. 3. State/mode indications from the RTE or other BSW modules such as the bus specific State Managers. Note that the BswM treats all request and indications in the exact same way.
BswMRule	0..*	Each instance of this container describes a BswM arbitration rule. The rule either consists of a simple mode condition or a more complex logical expression. This container also references the action lists that shall be invoked when the rule is evaluated to True or False.

⌋

10.2.4 BswMLogicalExpression

[ECUC_BswM_00808] Definition of EcucParamConfContainerDef BswMLogicalExpression ⌈

Container Name	BswMLogicalExpression
Parent Container	BswMArbitration
Upstream mapping defined	–
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.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMLogicalOperator	0..1	[ECUC_BswM_00814]
BswMArgumentRef	1..*	[ECUC_BswM_00820]

No Included Containers

⌋

[ECUC_BswM_00814] Definition of EcucEnumerationParamDef BswMLogicalOperator ⌈

Parameter Name	BswMLogicalOperator	
Parent Container	BswMLogicalExpression	
Upstream mapping defined	–	
Description	This parameter specifies the logical operator to be used in the logical expression. If the logical operator is set to something other than BSWM_NOT, and the expression only consists of a single condition, then this parameter will have no effect.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	BSWM_AND	And (&&)
	BSWM_NAND	Not And (NAND)
	BSWM_NOT	Not (!)
	BSWM_OR	Or (||)
	BSWM_XOR	Exclusive Or (XOR)
Pre-Compile only	true	

⌋

[ECUC_BswM_00820] Definition of EcucChoiceReferenceDef BswMArgumentRef ⌈

Parameter Name	BswMArgumentRef
Parent Container	BswMLogicalExpression
Upstream mapping defined	–

▽

△

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. In case the BswMLogicalExpression.BswMLogical Operator equals BSWM_NOT only one operand is supported.
Multiplicity	1..*
Type	Choice reference to [BswMLogicalExpression, BswMModeCondition]
Pre-Compile only	true

⌋

[SWS_BswM_00242]

Upstream requirements: SRS_BSW_00167, SRS_ModeMgm_02002

⌈The BswM shall reject configurations where a `BswMLogicalExpression` has a `BswMLogicalOperator` equal to BSWM_NAND and its number of `BswMArgument-Ref`s is not two.⌋

[SWS_BswM_00243]

Upstream requirements: SRS_BSW_00167, SRS_ModeMgm_02002

⌈The BswM shall reject configurations where a `BswMLogicalExpression` has a `BswMLogicalOperator` equal to BSWM_NOT and its number of `BswMArgument-Ref`s is not one.⌋

[SWS_BswM_00244]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall implement BSWM_XOR to evaluate to TRUE if an odd number of its arguments is TRUE, and evaluate to FALSE if an even number of its arguments is TRUE.⌋

[SWS_BswM_00245]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall implement BSWM_AND to evaluate to TRUE if all of its arguments are TRUE, and evaluate to FALSE if at least one of its arguments is FALSE.⌋

[SWS_BswM_00246]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall implement BSWM_NAND to evaluate to FALSE if all of its arguments are TRUE, and evaluate to TRUE if at least one of its arguments is FALSE.⌋

[SWS_BswM_00247]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall implement BSWM_OR to evaluate to FALSE if all of its arguments are FALSE, and evaluate to TRUE if at least one of its arguments is TRUE.⌋

[SWS_BswM_00248]

Upstream requirements: SRS_ModeMgm_02002

⌈The BswM shall implement BSWM_NOT to evaluate to FALSE if its argument is TRUE, and evaluate to TRUE if its argument is FALSE.⌋

10.2.5 BswMModeCondition

[ECUC_BswM_00807] Definition of EcucParamConfContainerDef BswMMode Condition ⌈

Container Name	BswMModeCondition
Parent Container	BswMArbitration
Upstream mapping defined	–
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.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMConditionType	1	[ECUC_BswM_00815]
BswMConditionMode	1	[ECUC_BswM_00821]

Included Containers		
Container Name	**Multiplicity**	**Description**
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.

⌋

[ECUC_BswM_00815] Definition of EcucEnumerationParamDef BswMCondition Type ⌈

Parameter Name	BswMConditionType
Parent Container	BswMModeCondition
Upstream mapping defined	–
Description	This parameter specifies what kind of comparison that is made for the evaluation of the mode condition. For BSWM_EQUALS and BSWM_EQUALS_NOT, the BswMModeRequestPort port referenced by BswMConditionMode is compared with the value configured in Bsw MConditionValue for equality or not-equality. For BSWM_EVENT_IS_SET and BSWM_EVENT_IS_CLEARED, the BswMEvent RequestPort port referenced by BswMConditionMode is checked for being set or cleared (not-set).
Multiplicity	1

▽

△

Type	EcucEnumerationParamDef	
Range	BSWM_EQUALS	Equals (=)
	BSWM_EQUALS_NOT	Not equals (!=)
	BSWM_EVENT_IS_CLEARED	Event is cleared
	BSWM_EVENT_IS_SET	Event is set
Pre-Compile only	true	

⌋

[ECUC_BswM_00821] Definition of EcucChoiceReferenceDef BswMCondition Mode ⌈

Parameter Name	BswMConditionMode
Parent Container	BswMModeCondition
Upstream mapping defined	–
Description	This parameter references either a mode request port or an event request port.
Multiplicity	1
Type	Choice reference to [BswMEventRequestPort, BswMModeRequestPort]
Pre-Compile only	true

⌋

[SWS_BswM_00256] Configuration constraint for mode request comparisons and event request checking

Upstream requirements: SRS_BSW_00167, SRS_ModeMgm_02001, SRS_ModeMgm_02002

⌈The BswM shall reject configurations where `BswMConditionType` BSWM_EQUALS or BSWM_EQUALS_NOT are configured in conjunction with a `BswMEvent-RequestPort` port referenced by the `BswMConditionMode`. The BswM shall reject configurations where `BswMConditionType` BSWM_EVENT_IS_SET or BSWM_EVENT_IS_CLEARED are configured in conjunction with a `BswMMode-RequestPort` port referenced by the `BswMConditionMode`.⌋

10.2.6 BswMConditionValue

[ECUC_BswM_00816] Definition of EcucChoiceContainerDef BswMCondition Value ⌈

Choice Container Name	BswMConditionValue
Parent Container	BswMModeCondition
Upstream mapping defined	–
Description	This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to.
Multiplicity	0..1
Pre-Compile only	true

No Included Parameters

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

⌋

10.2.7 BswMBswMode

[ECUC_BswM_00869] Definition of EcucParamConfContainerDef BswMBswMode ⌈

Container Name	BswMBswMode
Parent Container	BswMConditionValue
Upstream mapping defined	–
Description	This container defines the value of a mode in the BSW.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMBswRequestedMode	1	[ECUC_BswM_00866]

No Included Containers

⌋

[ECUC_BswM_00866] Definition of EcucStringParamDef BswMBswRequestedMode ⌈

Parameter Name	BswMBswRequestedMode
Parent Container	BswMBswMode
Upstream mapping defined	–
Description	This parameter contains the symbolic name (as a string) of a certain mode/state that can be requested/indicated by the BSW modules.
Multiplicity	1
Type	EcucStringParamDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8 BswMModeDeclaration

[ECUC_BswM_00868] Definition of EcucParamConfContainerDef BswMMode Declaration ⌈

Container Name	BswMModeDeclaration
Parent Container	BswMConditionValue
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMModeValueRef	1	[ECUC_BswM_00864]

No Included Containers

⌋

[ECUC_BswM_00864] Definition of EcucForeignReferenceDef BswMModeValue Ref ⌈

Parameter Name	BswMModeValueRef
Parent Container	BswMModeDeclaration
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
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
Pre-Compile only	true

⌋

10.2.9 BswMEventRequestPort

[ECUC_BswM_01052] Definition of EcucParamConfContainerDef BswMEventRequestPort ⌈

Container Name	BswMEventRequestPort
Parent Container	BswMArbitration
Upstream mapping defined	–

▽

△

Description	Each instance of this container defines an event which can be sent to the BswM. Basic Software Modules may send these events to the BswM by calling the corresponding BswM C-API (for example: BswM_ComM_InitiateReset()).
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEventRequestProcessing	1	[ECUC_BswM_01056]

Included Containers		
Container Name	**Multiplicity**	**Description**
BswMEventRequestSource	1	This choice container specifies the source of the event request. The sender of the event can be another BSW Module, such as ComM.

⌋

[ECUC_BswM_01056] Definition of EcucEnumerationParamDef BswMEventRequestProcessing ⌈

Parameter Name	BswMEventRequestProcessing	
Parent Container	BswMEventRequestPort	
Upstream mapping defined	–	
Description	This parameter defines if the processing of the mode arbitration shall be done immediately when an event request is received or if it shall be deferred to the processing of the main function of BswM.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_DEFERRED	Deferred
	BSWM_IMMEDIATE	Immediate
Pre-Compile only	true	

⌋

10.2.10 BswMModeRequestPort

[ECUC_BswM_00805] Definition of EcucParamConfContainerDef BswMModeRequestPort ⌈

Container Name	BswMModeRequestPort
Parent Container	BswMArbitration
Upstream mapping defined	–

▽

△

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.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMRequestProcessing	1	[ECUC_BswM_00822]

Included Containers		
Container Name	**Multiplicity**	**Description**
BswMModeInitValue	0..1	This container defines the initial mode value that is used by Bsw M for the corresponding mode request after initialization. The initial mode value is defined by configuring either BswMBsw ModeInitValue or BswMCompuScaleModeValue. This container is optional.
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.

⌋

[ECUC_BswM_00822] Definition of EcucEnumerationParamDef BswMRequest Processing ⌈

Parameter Name	BswMRequestProcessing	
Parent Container	BswMModeRequestPort	
Upstream mapping defined	–	
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	Deferred
	BSWM_IMMEDIATE	Immediate
Pre-Compile only	true	

⌋

10.2.11 BswMModeInitValue

[ECUC_BswM_00928] Definition of EcucParamConfContainerDef BswMModeInit Value ⌈

Container Name	BswMModeInitValue
Parent Container	BswMModeRequestPort
Upstream mapping defined	–
Description	This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. The initial mode value is defined by configuring either BswMBswModeInitValue or BswMCompuScaleModeValue. This container is optional.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMBswModeInitValue	0..1	[ECUC_BswM_00932]

Included Containers		
Container Name	**Multiplicity**	**Description**
BswMCompuScaleModeValue	0..1	This container contains parameters used to define a mode value.

⌋

[ECUC_BswM_00932] Definition of EcucStringParamDef BswMBswModeInitValue ⌈

Parameter Name	BswMBswModeInitValue
Parent Container	BswMModeInitValue
Upstream mapping defined	–
Description	This parameter defines the initial mode value that is used by BswM for the corresponding mode request after initialization.
Multiplicity	0..1
Type	EcucStringParamDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.12 BswMCompuScaleModeValue

[ECUC_BswM_01039] Definition of EcucParamConfContainerDef BswMCompuScaleModeValue ⌈

Container Name	BswMCompuScaleModeValue
Parent Container	BswMConditionValue, BswMModeInitValue
Upstream mapping defined	–
Description	This container contains parameters used to define a mode value.
Multiplicity	0..1

▽

△

Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMCompuConstText	1	[ECUC_BswM_01041]
BswMCompuMethodRef	1	[ECUC_BswM_01040]

No Included Containers

⌋

[ECUC_BswM_01041] Definition of EcucStringParamDef BswMCompuConstText ⌈

Parameter Name	BswMCompuConstText
Parent Container	BswMCompuScaleModeValue
Upstream mapping defined	–
Description	The value of this parameter shall match the VT member of a CompuConst defined within the referenced CompuMethod (BswMCompuMethodRef). The interval value of the corresponding CompuScale shall be used as the mode request value.
Multiplicity	1
Type	EcucStringParamDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_BswM_01040] Definition of EcucForeignReferenceDef BswMCompuMethodRef ⌈

Parameter Name	BswMCompuMethodRef
Parent Container	BswMCompuScaleModeValue
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is a foreign reference to the CompuMethod used for mode requests.
Multiplicity	1
Type	Foreign reference to COMPU-METHOD
Pre-Compile only	true

⌋

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

10.2.13 BswMEventRequestSource

[ECUC_BswM_01053] Definition of EcucChoiceContainerDef BswMEventRequestSource ⌈

Choice Container Name	BswMEventRequestSource
Parent Container	BswMEventRequestPort
Upstream mapping defined	–
Description	This choice container specifies the source of the event request. The sender of the event can be another BSW Module, such as ComM.
Multiplicity	1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMComMInitiateReset	0..1	This is an indication from the ComM to signal a shutdown.
BswMDcmApplicationUpdated Indication	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.
BswMNmCarWakeUpIndication	0..1	This is an indication of a CarWakeup from the Nm.
BswMSwitchAckNotification	0..1	This is a notification that a mode transition has been completed.

⌋

10.2.14 BswMModeRequestSource

[ECUC_BswM_00856] Definition of EcucChoiceContainerDef BswMModeRequestSource ⌈

Choice Container Name	BswMModeRequestSource
Parent Container	BswMModeRequestPort
Upstream mapping defined	–
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.
Multiplicity	1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMBswModeNotification	0..1	This is a mode request source emanating from another BSW Module.
BswMCanSMIndication	0..1	This is an indication of the current state of the CAN State Manager.

▽

△

Container Choices		
Container Name	**Multiplicity**	**Description**
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.
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.
BswMEcuMRUNRequestIndication	0..1	This is an indication of the current State of the RUN Request Protocol.
BswMEcuMWakeupSource	0..1	This is a notification of the current state of an ECU State Manager wakeup source.
BswMEthIfPortGroupLinkStateChg	0..1	This is an indication from the EthIf if the link state of a Ethernet interface switch port group has changed.
BswMEthSMIndication	0..1	This is an indication of the current state of the Ethernet State Manager.
BswMFrSMIndication	0..1	This is an indication of the current state of the FlexRay State Manager.
BswMGenericRequest	0..1	This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers).
BswMJ1939DcmBroadcastStatus	0..1	This is a notification of the desired broadcast status per network, triggered via DM13.
BswMJ1939NmIndication	0..1	This is an indication of the current state of the J1939 network management module.
BswMLinSMIndication	0..1	This is an indication of the current state of the LIN State Manager.
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.
BswMMkaControlledPortStatus Indication	0..1	This is an indication of the current state of a MACsec/CANsec controlled port
BswMNmStateChangeNotification	0..1	This is a notification from the Nm module that its state has changed.
BswMNvMJobModeIndication	0..1	Indicates the current status of the multiblock job. The job is identified via BswMNvmService. Possible values for this indication are the possible values of NvM_RequestResultType.
BswMNvMRequest	0..1	Via this Mode Request Source the NvM indicates the current status of the specified block. Possible Values are: NvM_Request ResultType 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
BswMSdClientServiceCurrentState	0..1	Used by Service Discovery module to indicate current state of the Client Service (available/down).
BswMSdConsumedEventGroup CurrentState	0..1	Used by Service Discovery to indicate current status of the Event Handler (requested/released).
BswMSdEventHandlerCurrentState	0..1	Used by Service Discovery to indicate current status of the Event Handler (requested/released).

▽

△

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMSoAdSoConModeChg	0..1	This is an indication of the current state of the SoAd.
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.
BswMTimer	0..1	This is a timer which can be used for time dependent rules. This mode request port can be in one of three modes (depending on the state of the timer): • BSWM_TIMER_STOPPED (intial) (The timer has been stopped by an action) • BSWM_TIMER_STARTED (The timer has been started by an action) • BSWM_TIMER_EXPIRED (The timer has expired)

⌋

10.2.15 BswMBswModeNotification

[ECUC_BswM_00926] Definition of EcucParamConfContainerDef BswMBswModeNotification ⌈

Container Name	BswMBswModeNotification
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a mode request source emanating from another BSW Module.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMBswModeDeclarationGroupPrototypeRef	1	[ECUC_BswM_00927]

No Included Containers

⌋

[ECUC_BswM_00927] Definition of EcucForeignReferenceDef BswMBswModeDeclarationGroupPrototypeRef ⌈

Parameter Name	BswMBswModeDeclarationGroupPrototypeRef
Parent Container	BswMBswModeNotification
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is a foreign reference to the Mode Declaration Group Prototype.
Multiplicity	1

▽

△

Type	Foreign reference to MODE-DECLARATION-GROUP-PROTOTYPE
Pre-Compile only	true

⌋

10.2.16 BswMCanSMIndication

[ECUC_BswM_00857] Definition of EcucParamConfContainerDef BswMCanSMIndication ⌈

Container Name	BswMCanSMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the CAN State Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMCanSMChannelRef	1	[ECUC_BswM_00870]

No Included Containers

⌋

[ECUC_BswM_00870] Definition of EcucReferenceDef BswMCanSMChannelRef ⌈

Parameter Name	BswMCanSMChannelRef
Parent Container	BswMCanSMIndication
Upstream mapping defined	–
Description	This is a reference to the CAN channel handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.17 BswMComMIndication

[ECUC_BswM_00880] Definition of EcucParamConfContainerDef BswMComMIndication ⌈

Container Name	BswMComMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current communication mode of a channel in the Communication Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMComMChannelRef	1	[ECUC_BswM_00883]

No Included Containers

⌋

[ECUC_BswM_00883] Definition of EcucReferenceDef BswMComMChannelRef ⌈

Parameter Name	BswMComMChannelRef
Parent Container	BswMComMIndication
Upstream mapping defined	–
Description	This is a reference to the Communication Manager channel handle that the indication corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.18 BswMComMInitiateReset

[ECUC_BswM_01014] Definition of EcucParamConfContainerDef BswMComMInitiateReset ⌈

Container Name	BswMComMInitiateReset
Parent Container	BswMEventRequestSource
Upstream mapping defined	–
Description	This is an indication from the ComM to signal a shutdown.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

No Included Containers

⌋

10.2.19 BswMComMPncRequest

[ECUC_BswM_00922] Definition of EcucParamConfContainerDef BswMComMPncRequest ⌈

Container Name	BswMComMPncRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMComMPncRef	1	[ECUC_BswM_00924]

No Included Containers

⌋

[ECUC_BswM_00924] Definition of EcucReferenceDef BswMComMPncRef ⌈

Parameter Name	BswMComMPncRef
Parent Container	BswMComMPncRequest
Upstream mapping defined	–
Description	This is a reference to the Communication Manager PNC handle of the Partial Network Cluster that the request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMPnc
Pre-Compile only	true

⌋

10.2.20 BswMDcmApplicationUpdatedIndication

[ECUC_BswM_00925] Definition of EcucParamConfContainerDef BswMDcmApplicationUpdatedIndication ⌈

Container Name	BswMDcmApplicationUpdatedIndication
Parent Container	BswMEventRequestSource
Upstream mapping defined	–
Description	This is a request to update application data from the DCM. This container does not contain any parameters since there are no further configuration needed for this type of request.
Multiplicity	0..1

▽

△

Pre-Compile only	true
Configuration Parameters	

No Included Parameters

No Included Containers

⌋

10.2.21 BswMDcmComModeRequest

[ECUC_BswM_00863] Definition of EcucParamConfContainerDef BswMDcmComModeRequest ⌈

Container Name	BswMDcmComModeRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	The source of the mode request is the Diagnostic Communication Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMDcmComMChannelRef	1	[ECUC_BswM_00969]

No Included Containers

⌋

[ECUC_BswM_00969] Definition of EcucReferenceDef BswMDcmComMChannelRef ⌈

Parameter Name	BswMDcmComMChannelRef
Parent Container	BswMDcmComModeRequest
Upstream mapping defined	–
Description	This is a reference from DcmModeRequest to the ComM channel that the indication corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.22 BswMEcuMIndication

[ECUC_BswM_01085] Definition of EcucParamConfContainerDef BswMEcuMIndication ⌈

Container Name	BswMEcuMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

No Included Containers

⌋

10.2.23 BswMEcuMRUNRequestIndication

[ECUC_BswM_01086] Definition of EcucParamConfContainerDef BswMEcuMRUNRequestIndication ⌈

Container Name	BswMEcuMRUNRequestIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current State of the RUN Request Protocol.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEcuMRUNRequestProtocolPort	1	[ECUC_BswM_01087]

No Included Containers

⌋

[ECUC_BswM_01087] Definition of EcucEnumerationParamDef BswMEcuMRUN-RequestProtocolPort ⌈

Parameter Name	BswMEcuMRUNRequestProtocolPort	
Parent Container	BswMEcuMRUNRequestIndication	
Upstream mapping defined	–	
Description	Identifies the EcuM State which is related to the mode request.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_ECUM_STATE_POST_RUN	Port for POST_RUN State of EcuM.
	BSWM_ECUM_STATE_RUN	Port for RUN State of EcuM.
Pre-Compile only	true	

⌋

10.2.24 BswMEcuMWakeupSource

[ECUC_BswM_00904] Definition of EcucParamConfContainerDef BswMEcuMWakeupSource ⌈

Container Name	BswMEcuMWakeupSource
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a notification of the current state of an ECU State Manager wakeup source.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEcuMWakeupSrcRef	1	[ECUC_BswM_00905]

No Included Containers

⌋

[ECUC_BswM_00905] Definition of EcucReferenceDef BswMEcuMWakeupSrcRef ⌈

Parameter Name	BswMEcuMWakeupSrcRef
Parent Container	BswMEcuMWakeupSource
Upstream mapping defined	–
Description	This is a reference to the ECU State Manager Wakeup Source that the indication corresponds to.
Multiplicity	1
Type	Symbolic name reference to EcuMWakeupSource

▽

△

Pre-Compile only	true

⌋

10.2.25 BswMEthIfPortGroupLinkStateChg

[ECUC_BswM_01066] Definition of EcucParamConfContainerDef BswMEthIfPortGroupLinkStateChg ⌈

Container Name	BswMEthIfPortGroupLinkStateChg
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication from the EthIf if the link state of a Ethernet interface switch port group has changed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEthIfSwitchPortGroupRef	1	[ECUC_BswM_01067]

No Included Containers

⌋

[ECUC_BswM_01067] Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef ⌈

Parameter Name	BswMEthIfSwitchPortGroupRef	
Parent Container	BswMEthIfPortGroupLinkStateChg	
Upstream mapping defined	–	
Description	This is a reference to the Ethernet Interface Switch Port Group that the indication corresponds to.	
Multiplicity	1	
Type	Symbolic name reference to EthIfSwitchPortGroup	
Post-Build Value	-	Selectable

⌋

10.2.26 BswMEthSMIndication

[ECUC_BswM_00860] Definition of EcucParamConfContainerDef BswMEthSMIndication ⌈

Container Name	BswMEthSMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the Ethernet State Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMEthSMChannelRef	1	[ECUC_BswM_00873]

No Included Containers

⌋

[ECUC_BswM_00873] Definition of EcucReferenceDef BswMEthSMChannelRef ⌈

Parameter Name	BswMEthSMChannelRef
Parent Container	BswMEthSMIndication
Upstream mapping defined	–
Description	This is a reference to the Ethernet channel handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.27 BswMFrSMIndication

[ECUC_BswM_00858] Definition of EcucParamConfContainerDef BswMFrSMIndication ⌈

Container Name	BswMFrSMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the FlexRay State Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMFrSMChannelRef	1	[ECUC_BswM_00872]

No Included Containers

⌋

[ECUC_BswM_00872] Definition of EcucReferenceDef BswMFrSMChannelRef ⌈

Parameter Name	BswMFrSMChannelRef
Parent Container	BswMFrSMIndication
Upstream mapping defined	–
Description	This is a reference to the FlexRay Cluster handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.28 BswMGenericRequest

[ECUC_BswM_00861] Definition of EcucParamConfContainerDef BswMGeneric Request ⌈

Container Name	BswMGenericRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers).
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMModeRequesterId	1	[ECUC_BswM_00874]

No Included Containers

⌋

[ECUC_BswM_00874] Definition of EcucIntegerParamDef BswMModeRequester Id ⌈

Parameter Name	BswMModeRequesterId	
Parent Container	BswMGenericRequest	
Upstream mapping defined	–	
Description	This parameters identifies the different users of the generic mode request interface.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	

▽

△

Pre-Compile only	true

⌋

10.2.29 BswMJ1939DcmBroadcastStatus

[ECUC_BswM_00985] Definition of EcucParamConfContainerDef BswMJ1939DcmBroadcastStatus ⌈

Container Name	BswMJ1939DcmBroadcastStatus
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a notification of the desired broadcast status per network, triggered via DM13.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMJ1939DcmChannelRef	1	[ECUC_BswM_00988]

No Included Containers

⌋

[ECUC_BswM_00988] Definition of EcucReferenceDef BswMJ1939DcmChannelRef ⌈

Parameter Name	BswMJ1939DcmChannelRef
Parent Container	BswMJ1939DcmBroadcastStatus
Upstream mapping defined	–
Description	Reference to the communication channel which is affected by this mode request.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

[SWS_BswM_00250]

Upstream requirements: SRS_ModeMgm_02001

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

10.2.30 BswMJ1939NmIndication

[ECUC_BswM_00966] Definition of EcucParamConfContainerDef BswMJ1939Nm Indication ⌈

Container Name	BswMJ1939NmIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the J1939 network management module.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMJ1939NmChannelRef	1	[ECUC_BswM_00967]
BswMJ1939NmNodeRef	1	[ECUC_BswM_00997]

No Included Containers

⌋

[ECUC_BswM_00967] Definition of EcucReferenceDef BswMJ1939NmChannel Ref ⌈

Parameter Name	BswMJ1939NmChannelRef
Parent Container	BswMJ1939NmIndication
Upstream mapping defined	–
Description	This is a reference to the J1939Nm channel handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

[ECUC_BswM_00997] Definition of EcucReferenceDef BswMJ1939NmNodeRef ⌈

Parameter Name	BswMJ1939NmNodeRef
Parent Container	BswMJ1939NmIndication
Upstream mapping defined	–
Description	This is a reference to the node that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to J1939NmNode
Pre-Compile only	true

⌋

10.2.31 BswMLinSMIndication

[ECUC_BswM_00859] Definition of EcucParamConfContainerDef BswMLinSMIndication ⌈

Container Name	BswMLinSMIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the LIN State Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMLinSMChannelRef	1	[ECUC_BswM_00871]

No Included Containers

⌋

[ECUC_BswM_00871] Definition of EcucReferenceDef BswMLinSMChannelRef ⌈

Parameter Name	BswMLinSMChannelRef
Parent Container	BswMLinSMIndication
Upstream mapping defined	–
Description	This is a reference to the LIN channel handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.32 BswMLinScheduleIndication

[ECUC_BswM_00885] Definition of EcucParamConfContainerDef BswMLinScheduleIndication ⌈

Container Name	BswMLinScheduleIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the currently active LIN Schedule Table for a specific LIN Interface.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMLinSMChannelRef	1	[ECUC_BswM_01028]

No Included Containers

⌋

[ECUC_BswM_01028] Definition of EcucReferenceDef BswMLinSMChannelRef ⌈

Parameter Name	BswMLinSMChannelRef	
Parent Container	BswMLinScheduleIndication	
Upstream mapping defined	–	
Description	This is a reference to the LIN channel handle that the mode request corresponds to.	
Multiplicity	1	
Type	Symbolic name reference to ComMChannel	
Post-Build Value	Loadable	Selectable

⌋

10.2.33 BswMLinTpModeRequest

[ECUC_BswM_00914] Definition of EcucParamConfContainerDef BswMLinTpModeRequest ⌈

Container Name	BswMLinTpModeRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMLinTpChannelRef	1	[ECUC_BswM_00915]

No Included Containers

⌋

[ECUC_BswM_00915] Definition of EcucReferenceDef BswMLinTpChannelRef ⌈

Parameter Name	BswMLinTpChannelRef
Parent Container	BswMLinTpModeRequest
Upstream mapping defined	–

▽

△

Description	This is a reference to the LIN Interface Channel that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.34 BswMMkaControlledPortStatusIndication

[ECUC_BswM_01098] Definition of EcucParamConfContainerDef BswMMkaControlledPortStatusIndication ⌈

Container Name	BswMMkaControlledPortStatusIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of a MACsec/CANsec controlled port
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMMkaChannelRef	1	[ECUC_BswM_01099]

No Included Containers

⌋

[ECUC_BswM_01099] Definition of EcucReferenceDef BswMMkaChannelRef ⌈

Parameter Name	BswMMkaChannelRef
Parent Container	BswMMkaControlledPortStatusIndication
Upstream mapping defined	–
Description	This is a reference to the CAN/Ethernet channel handle that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.35 BswMNmCarWakeUpIndication

[ECUC_BswM_01075] Definition of EcucParamConfContainerDef BswMNmCarWakeUpIndication ⌈

Container Name	BswMNmCarWakeUpIndication
Parent Container	BswMEventRequestSource
Upstream mapping defined	–
Description	This is an indication of a CarWakeup from the Nm.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMNmChannelRef	1	[ECUC_BswM_01049]

No Included Containers

⌋

[ECUC_BswM_01049] Definition of EcucReferenceDef BswMNmChannelRef ⌈

Parameter Name	BswMNmChannelRef
Parent Container	BswMNmCarWakeUpIndication
Upstream mapping defined	–
Description	This is a reference to the channel handle that the indication corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.36 BswMNmStateChangeNotification

[ECUC_BswM_01081] Definition of EcucParamConfContainerDef BswMNmStateChangeNotification ⌈

Container Name	BswMNmStateChangeNotification
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a notification from the Nm module that its state has changed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMNmChannelRef	1	[ECUC_BswM_01082]

No Included Containers

⌋

[ECUC_BswM_01082] Definition of EcucReferenceDef BswMNmChannelRef ⌈

Parameter Name	BswMNmChannelRef
Parent Container	BswMNmStateChangeNotification
Upstream mapping defined	–
Description	This is a reference to the channel handle that this notification corresponds to.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.37 BswMNvMJobModeIndication

[ECUC_BswM_00956] Definition of EcucParamConfContainerDef BswMNvMJob ModeIndication ⌈

Container Name	BswMNvMJobModeIndication
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	Indicates the current status of the multiblock job. The job is identified via BswMNvm Service. Possible values for this indication are the possible values of NvM_Request ResultType.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMNvmService	1	[ECUC_BswM_00957]

No Included Containers

⌋

[ECUC_BswM_00957] Definition of EcucEnumerationParamDef BswMNvmService ⌈

Parameter Name	BswMNvmService	
Parent Container	BswMNvMJobModeIndication	
Upstream mapping defined	–	
Description	Identifies the Nvm job which is related to the mode request.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	NvmCancelWriteAll	corresponds to multi block service NvM_Cancel WriteAll

▽

△

	NvmFirstInitAll	corresponds to multi block service NvM_FirstInit All
	NvmReadAll	corresponds to multi block service NvM_ReadAll
	NvmValidateAll	corresponds to multi block service NvM_Validate All
	NvmWriteAll	corresponds to multi block service NvM_WriteAll
Pre-Compile only	true	

⌋

10.2.38 BswMNvMRequest

[ECUC_BswM_00890] Definition of EcucParamConfContainerDef BswMNvMRequest ⌈

Container Name	BswMNvMRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
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
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMNvMBlockRef	1	[ECUC_BswM_00891]

No Included Containers

⌋

[ECUC_BswM_00891] Definition of EcucReferenceDef BswMNvMBlockRef ⌈

Parameter Name	BswMNvMBlockRef
Parent Container	BswMNvMRequest
Upstream mapping defined	–
Description	This is a reference to the NvM Block Descriptor that the request corresponds to.
Multiplicity	1
Type	Symbolic name reference to NvMBlockDescriptor
Pre-Compile only	true

⌋

10.2.39 BswMSdClientServiceCurrentState

[ECUC_BswM_01011] Definition of EcucParamConfContainerDef BswMSdClient ServiceCurrentState ⌈

Container Name	BswMSdClientServiceCurrentState
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	Used by Service Discovery module to indicate current state of the Client Service (available/down).
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdClientMethodsRef	1	[ECUC_BswM_01009]

No Included Containers

⌋

[ECUC_BswM_01009] Definition of EcucReferenceDef BswMSdClientMethods Ref ⌈

Parameter Name	BswMSdClientMethodsRef
Parent Container	BswMSdClientServiceCurrentState, BswMSdClientServiceModeRequest
Upstream mapping defined	–
Description	This is a reference to a client service in the Sd module.
Multiplicity	1
Type	Symbolic name reference to SdClientService
Pre-Compile only	true

⌋

10.2.40 BswMSdConsumedEventGroupCurrentState

[ECUC_BswM_01012] Definition of EcucParamConfContainerDef BswMSdConsumedEventGroupCurrentState ⌈

Container Name	BswMSdConsumedEventGroupCurrentState
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	Used by Service Discovery to indicate current status of the EventHandler (requested/ released).
Multiplicity	0..1

▽

△

Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdConsumedEventGroupRef	1	[ECUC_BswM_01010]

No Included Containers

⌋

[ECUC_BswM_01010] Definition of EcucReferenceDef BswMSdConsumedEvent GroupRef ⌈

Parameter Name	BswMSdConsumedEventGroupRef
Parent Container	BswMSdConsumedEventGroupCurrentState, BswMSdConsumedEventGroupMode Request
Upstream mapping defined	–
Description	This is a reference to an eventGroup that is defined within a client service in the Sd module.
Multiplicity	1
Type	Symbolic name reference to SdConsumedEventGroup
Pre-Compile only	true

⌋

10.2.41 BswMSdEventHandlerCurrentState

[ECUC_BswM_01013] Definition of EcucParamConfContainerDef BswMSdEvent HandlerCurrentState ⌈

Container Name	BswMSdEventHandlerCurrentState
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	Used by Service Discovery to indicate current status of the EventHandler (requested/ released).
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdEventHandlerRef	1	[ECUC_BswM_01008]

No Included Containers

⌋

[ECUC_BswM_01008] Definition of EcucReferenceDef BswMSdEventHandlerRef ⌈

Parameter Name	BswMSdEventHandlerRef
Parent Container	BswMSdEventHandlerCurrentState
Upstream mapping defined	–
Description	This is a reference to an event handler that is defined within a server service in the Sd module.
Multiplicity	1
Type	Symbolic name reference to SdEventHandler
Pre-Compile only	true

⌋

10.2.42 BswMSoAdSoConModeChg

[ECUC_BswM_01091] Definition of EcucParamConfContainerDef BswMSoAdSoConModeChg ⌈

Container Name	BswMSoAdSoConModeChg
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is an indication of the current state of the SoAd.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSoAdSocketConnectionRef	1	[ECUC_BswM_01092]

No Included Containers

⌋

[ECUC_BswM_01092] Definition of EcucReferenceDef BswMSoAdSocketConnectionRef ⌈

Parameter Name	BswMSoAdSocketConnectionRef
Parent Container	BswMSoAdSoConModeChg
Upstream mapping defined	–
Description	This is a reference to the SoAdSocketConnection that the mode request corresponds to.
Multiplicity	1
Type	Symbolic name reference to SoAdSocketConnection
Pre-Compile only	true

⌋

10.2.43 BswMSwcModeNotification

[ECUC_BswM_00892] Definition of EcucParamConfContainerDef BswMSwcModeNotification ⌈

Container Name	BswMSwcModeNotification
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a mode switch notification associated with a RTE switch interface.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSwcModeNotificationModeDeclarationGroupPrototypeRef	1	[ECUC_BswM_00893]

No Included Containers

⌋

[ECUC_BswM_00893] Definition of EcucForeignReferenceDef BswMSwcModeNotificationModeDeclarationGroupPrototypeRef ⌈

Parameter Name	BswMSwcModeNotificationModeDeclarationGroupPrototypeRef
Parent Container	BswMSwcModeNotification
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is a foreign reference to the ModeDeclarationGroupPrototype.
Multiplicity	1
Type	Foreign reference to MODE-DECLARATION-GROUP-PROTOTYPE
Pre-Compile only	true

⌋

10.2.44 BswMSwcModeRequest

[ECUC_BswM_00862] Definition of EcucParamConfContainerDef BswMSwcModeRequest ⌈

Container Name	BswMSwcModeRequest
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	The source of the mode request is a SW Component.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSwcModeRequestVariableDataPrototypeRef	1	[ECUC_BswM_01046]

No Included Containers

⌋

[ECUC_BswM_01046] Definition of EcucForeignReferenceDef BswMSwcModeRequestVariableDataPrototypeRef ⌈

Parameter Name	BswMSwcModeRequestVariableDataPrototypeRef	
Parent Container	BswMSwcModeRequest	
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]	
Description	This is a reference to the VariableDataPrototype.	
Multiplicity	1	
Type	Foreign reference to VARIABLE-DATA-PROTOTYPE	
Post-Build Value	Loadable	Selectable

⌋

[SWS_BswM_00236]

Upstream requirements: SRS_ModeMgm_02001

⌈The BswM shall only accept configurations where the VariableDataPrototype, which is referenced by a `BswMSwcModeRequest`, belongs to a SenderReceiverInterface.⌋

10.2.45 BswMSwitchAckNotification

[ECUC_BswM_01083] Definition of EcucParamConfContainerDef BswMSwitchAckNotification ⌈

Container Name	BswMSwitchAckNotification
Parent Container	BswMEventRequestSource
Upstream mapping defined	–
Description	This is a notification that a mode transition has been completed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSwitchPortRef	1	[ECUC_BswM_01084]

No Included Containers

⌋

[ECUC_BswM_01084] Definition of EcucReferenceDef BswMSwitchPortRef ⌈

Parameter Name	BswMSwitchPortRef
Parent Container	BswMSwitchAckNotification
Upstream mapping defined	–
Description	References the switch port which will receive the notification.
Multiplicity	1
Type	Reference to BswMSwitchPort
Pre-Compile only	true

⌋

[SWS_BswM_00284]

Upstream requirements: SRS_ModeMgm_02001

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

[SWS_BswM_00285]

Upstream requirements: SRS_ModeMgm_02001

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

10.2.46 BswMTimer

[ECUC_BswM_01058] Definition of EcucParamConfContainerDef BswMTimer ⌈

Container Name	BswMTimer
Parent Container	BswMModeRequestSource
Upstream mapping defined	–
Description	This is a timer which can be used for time dependent rules. This mode request port can be in one of three modes (depending on the state of the timer): • BSWM_TIMER_STOPPED (intial) (The timer has been stopped by an action) • BSWM_TIMER_STARTED (The timer has been started by an action) • BSWM_TIMER_EXPIRED (The timer has expired)
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

No Included Containers

⌋

10.2.47 BswMRule

[ECUC_BswM_00806] Definition of EcucParamConfContainerDef BswMRule ⌈

Container Name	BswMRule	
Parent Container	BswMArbitration	
Upstream mapping defined	–	
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.	
Multiplicity	0..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMNestedExecutionOnly	1	[ECUC_BswM_00935]
BswMRuleInitState	1	[ECUC_BswM_00888]
BswMRuleExpressionRef	1	[ECUC_BswM_00819]
BswMRuleFalseActionList	0..1	[ECUC_BswM_00818]
BswMRuleTrueActionList	0..1	[ECUC_BswM_00817]

No Included Containers

⌋

[ECUC_BswM_00935] Definition of EcucBooleanParamDef BswMNestedExecutionOnly ⌈

Parameter Name	BswMNestedExecutionOnly
Parent Container	BswMRule
Upstream mapping defined	–
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
Pre-Compile only	true

⌋

[ECUC_BswM_00888] Definition of EcucEnumerationParamDef BswMRuleInitState ⌈

Parameter Name	BswMRuleInitState	
Parent Container	BswMRule	
Upstream mapping defined	–	
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	False
	BSWM_TRUE	True
	BSWM_UNDEFINED	Undefined
Pre-Compile only	true	

⌋

[ECUC_BswM_00819] Definition of EcucReferenceDef BswMRuleExpressionRef ⌈

Parameter Name	BswMRuleExpressionRef
Parent Container	BswMRule
Upstream mapping defined	–
Description	This is a reference to the logical expression that is evaluated for each rule.
Multiplicity	1
Type	Reference to BswMLogicalExpression
Pre-Compile only	true

⌋

[ECUC_BswM_00818] Definition of EcucReferenceDef BswMRuleFalseActionList ⌈

Parameter Name	BswMRuleFalseActionList	
Parent Container	BswMRule	
Upstream mapping defined	–	
Description	This is a reference to the action list that shall be executed when the rule is evaluated to False	
Multiplicity	0..1	
Type	Reference to BswMActionList	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00817] Definition of EcucReferenceDef BswMRuleTrueActionList ⌈

Parameter Name	BswMRuleTrueActionList	
Parent Container	BswMRule	
Upstream mapping defined	–	
Description	This is a reference to the action list that shall be executed when the rule is evaluated to True	
Multiplicity	0..1	
Type	Reference to BswMActionList	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

10.2.48 BswMDataTypeMappingSets

[ECUC_BswM_00936] Definition of EcucParamConfContainerDef BswMDataTypeMappingSets ⌈

Container Name	BswMDataTypeMappingSets
Parent Container	BswMConfig
Upstream mapping defined	–
Description	Collection of references to DataTypeMappingSet.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMDataTypeMappingSetRef	1..*	[ECUC_BswM_00937]

No Included Containers

⌋

[ECUC_BswM_00937] Definition of EcucForeignReferenceDef BswMDataTypeMappingSetRef ⌈

Parameter Name	BswMDataTypeMappingSetRef
Parent Container	BswMDataTypeMappingSets
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	Reference to DataTypeMappingSet.
Multiplicity	1..*
Type	Foreign reference to DATA-TYPE-MAPPING-SET
Pre-Compile only	true

⌋

10.2.49 BswMModeControl

[ECUC_BswM_00802] Definition of EcucParamConfContainerDef BswMMode Control ⌈

Container Name	BswMModeControl
Parent Container	BswMConfig
Upstream mapping defined	–
Description	This container includes all configuration sub-containers and parameters related to the mode control functionality of the BswM.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

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

⌋

10.2.50 BswMAction

[ECUC_BswM_00810] Definition of EcucParamConfContainerDef BswMAction ⌈

Container Name	BswMAction
Parent Container	BswMModeControl
Upstream mapping defined	–
Description	Each container of this type defines an action. These actions can be part of one or several action lists.

▽

△

Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

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

⌋

10.2.51 BswMAvailableActions

[ECUC_BswM_00826] Definition of EcucChoiceContainerDef BswMAvailableActions ⌈

Choice Container Name	BswMAvailableActions
Parent Container	BswMAction
Upstream mapping defined	–
Description	Choice container including the available actions to be used in the action lists.
Multiplicity	1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMClearEventRequest	0..1	This container contains a reference to a BswMEventRequestPort which will be cleared (i.e. set to CLEAR state) when this action is executed.
BswMComMAllowCom	0..1	This container includes all parameters for the action to allow or to block communication for a ComM Channel. ComM_ CommunicationAllowed is called when this action is configured.
BswMComMModeLimitation	0..1	This container includes all parameters related to a limitation of communication mode for a ComM Channel. ComM_Limit ChannelToNoComMode 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_RequestCom Mode 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.
BswMEcuMDriverInitListBswM	0..1	This container defines the action to trigger an EcuM driver initialization list.
BswMEcuMGoDownHaltPoll	0..1	This container defines the action to trigger the EcuM_GoDown HaltPoll from BswM.
BswMEcuMSelectShutdownTarget	0..1	This container defines the shutdown target.

▽

△

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMEcuMStateSwitch	0..1	This container defines the action to switch a State of the ECU State Manager.
BswMEthIfStartAllPorts	0..1	This container defines the action to call EthIf_StartAllPorts from BswM and thus to start all affected EthSwtPorts.
BswMEthIfSwitchPortGroup RequestMode	0..1	This container includes all parameters related to requesting a mode for the EthIfSwtPortGroup. The EthIf_SwitchPortGroupRequestMode API is called when this action is executed.
BswMFrSMAllSlots	0..1	This container includes all parameter(s) for the action to request an exit from Flexray KeySlotOnlyMode. FrSM_AllSlots is called when this action is executed.
BswMIdsMBlockStateChange Request	0..1	This container includes all parameters related to switching the Block State of the IdsM. The IdsM_BswM_StateChanged API is called when this action is executed. **Tags:** atp.Status=draft
BswMIdsMTransmissionState ChangeRequest	0..1	This container defines the action to switch the transmission state of the IdsM. The IdsM_TransmissionSetState API is called when this action is executed.
BswMJ1939DcmStateSwitch	0..1	This container includes all parameters related to a switch of the J1939 Diagnostic Communication Managers network state for a J1939 node. J1939Dcm_SetState is called when this action is configured.
BswMJ1939RmStateSwitch	0..1	This container includes all parameters related to a switch of the J1939 Request Managers network state for a J1939 node. J1939Rm_SetState is called when this action is configured.
BswMLinScheduleSwitch	0..1	This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured. 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_Enable Communication or Nm_DisableCommunication is called when this action is configured.
BswMPduGroupSwitch	0..1	This container includes references to the PDU groups that shall be enabled and disabled.
BswMPduRouterControl	0..1	This container includes all parameters related to enabling and disabling of routing of Routing Path Groups in the PDU Router. PduR_EnableRouting or PduR_DisableRouting is called when this action is configured.
BswMRteModeRequest	0..1	This container defines a mode request that the BswM may send to a SW-C which is acting as a mode-manager. RTE_Write is called when this action is configured.
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.

▽

△

Container Choices		
Container Name	**Multiplicity**	**Description**
BswMSdClientServiceMode Request	0..1	This container includes all parameters related to the selection of an client service of Sd. Sd_ClientServiceSetState is called when this action is configured.
BswMSdConsumedEventGroup ModeRequest	0..1	This container includes all parameters related to the selection of a consumed EventGroup of Sd. Sd_ConsumedEventGroupSet State is called when this action is configured.
BswMSdServerServiceMode Request	0..1	This container includes all parameters related to the selection of a server service of Sd. Sd_ServerServiceSetState is called when this action is configured.
BswMSdServiceGroupSwitch	0..1	This container includes references to the SdServiceGroups that shall be enabled and disabled.
BswMSetFirewallState	0..1	This container includes parameters for switching the state of the Firewall. The BswM will call Fw_SetFirewallState() when this action is executed. **Tags:** atp.Status=draft
BswMSwitchIPduMode	0..1	This container includes all parameters related to the selection of the transmission mode an I-PDU to be sent by COM. Com_ SwitchIpduTxMode is called when this action is configured.
BswMTimerControl	0..1	This container includes all parameters for the action to start or to stop a timer.
BswMTriggerIPduSend	0..1	This container includes all parameters related to the triggering of an I-PDU to be sent by COM. Com_TriggerIPDUSend is called when this action is configured.
BswMUserCallout	0..1	This container includes all details needed for a user defined function call.

⌋

10.2.52 BswMClearEventRequest

[ECUC_BswM_01054] Definition of EcucParamConfContainerDef BswMClear EventRequest ⌈

Container Name	BswMClearEventRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container contains a reference to a BswMEventRequestPort which will be cleared (i.e. set to CLEAR state) when this action is executed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMClearEventRequestPortRef	1	[ECUC_BswM_01055]

No Included Containers

⌋

[ECUC_BswM_01055] Definition of EcucReferenceDef BswMClearEventRequestPortRef ⌈

Parameter Name	BswMClearEventRequestPortRef
Parent Container	BswMClearEventRequest
Upstream mapping defined	–
Description	This parameter references the BswMEventRequestPort which will be cleared.
Multiplicity	1
Type	Reference to BswMEventRequestPort
Pre-Compile only	true

⌋

10.2.53 BswMComMAllowCom

[ECUC_BswM_00909] Definition of EcucParamConfContainerDef BswMComMAllowCom ⌈

Container Name	BswMComMAllowCom
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMComAllowed	1	[ECUC_BswM_00918]
BswMComMAllowChannelRef	1	[ECUC_BswM_00912]

No Included Containers

⌋

[ECUC_BswM_00918] Definition of EcucBooleanParamDef BswMComAllowed ⌈

Parameter Name	BswMComAllowed
Parent Container	BswMComMAllowCom
Upstream mapping defined	–

▽

△

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	–
Pre-Compile only	true

⌋

[ECUC_BswM_00912] Definition of EcucReferenceDef BswMComMAllowChannelRef ⌈

Parameter Name	BswMComMAllowChannelRef
Parent Container	BswMComMAllowCom
Upstream mapping defined	–
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	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.54 BswMComMModeLimitation

[ECUC_BswM_00908] Definition of EcucParamConfContainerDef BswMComMModeLimitation ⌈

Container Name	BswMComMModeLimitation
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMComMLimitMode	1	[ECUC_BswM_00910]
BswMComMLimitChannelRef	1	[ECUC_BswM_00911]

No Included Containers

⌋

[ECUC_BswM_00910] Definition of EcucBooleanParamDef BswMComMLimitMode ⌈

Parameter Name	BswMComMLimitMode
Parent Container	BswMComMModeLimitation
Upstream mapping defined	–
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	–
Pre-Compile only	true

⌋

[ECUC_BswM_00911] Definition of EcucReferenceDef BswMComMLimitChannelRef ⌈

Parameter Name	BswMComMLimitChannelRef
Parent Container	BswMComMModeLimitation
Upstream mapping defined	–
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	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.55 BswMComMModeSwitch

[ECUC_BswM_00831] Definition of EcucParamConfContainerDef BswMComMModeSwitch ⌈

Container Name	BswMComMModeSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMComMRequestedMode	1	[ECUC_BswM_00840]
BswMComMUserRef	1	[ECUC_BswM_00841]

No Included Containers

⌋

[ECUC_BswM_00840] Definition of EcucEnumerationParamDef BswMComMRequestedMode ⌈

Parameter Name	BswMComMRequestedMode	
Parent Container	BswMComMModeSwitch	
Upstream mapping defined	–	
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_COMM_FULL_COMMUNICATION	Request full communication
	BSWM_COMM_NO_COMMUNICATION	Request no communication
Pre-Compile only	true	

⌋

[ECUC_BswM_00841] Definition of EcucReferenceDef BswMComMUserRef ⌈

Parameter Name	BswMComMUserRef
Parent Container	BswMComMModeSwitch
Upstream mapping defined	–
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_Request ComMode.
Multiplicity	1
Type	Symbolic name reference to ComMUser
Pre-Compile only	true

⌋

10.2.56 BswMDeadlineMonitoringControl

[ECUC_BswM_00830] Definition of EcucParamConfContainerDef BswMDeadlineMonitoringControl ⌈

Container Name	BswMDeadlineMonitoringControl
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMDisabledDMPduGroupRef	0..*	[ECUC_BswM_00852]
BswMEnabledDMPduGroupRef	0..*	[ECUC_BswM_00851]

No Included Containers

⌋

[ECUC_BswM_00852] Definition of EcucReferenceDef BswMDisabledDMPduGroupRef ⌈

Parameter Name	BswMDisabledDMPduGroupRef	
Parent Container	BswMDeadlineMonitoringControl	
Upstream mapping defined	–	
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be disabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_DisableReceptionDM.	
Multiplicity	0..*	
Type	Symbolic name reference to ComIPduGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00851] Definition of EcucReferenceDef BswMEnabledDMPduGroupRef ⌈

Parameter Name	BswMEnabledDMPduGroupRef	
Parent Container	BswMDeadlineMonitoringControl	
Upstream mapping defined	–	
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be enabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_EnableReceptionDM.	
Multiplicity	0..*	
Type	Symbolic name reference to ComIPduGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

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

[SWS_BswM_00274]

Upstream requirements: SRS_ModeMgm_02007, SRS_ModeMgm_02003

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

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

10.2.57 BswMEcuMDriverInitListBswM

[ECUC_BswM_01064] Definition of EcucParamConfContainerDef BswMEcuMDriverInitListBswM ⌈

Container Name	BswMEcuMDriverInitListBswM
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the action to trigger an EcuM driver initialization list.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEcuMDriverInitListBswMRef	1	[ECUC_BswM_01065]

No Included Containers

⌋

[ECUC_BswM_01065] Definition of EcucReferenceDef BswMEcuMDriverInitListBswMRef ⌈

Parameter Name	BswMEcuMDriverInitListBswMRef
Parent Container	BswMEcuMDriverInitListBswM
Upstream mapping defined	–
Description	This is a reference to the EcuM EcuMDriverInitListBswM container which represents the driver init list to be triggered.
Multiplicity	1

▽

△

Type	Reference to EcuMDriverInitListBswM
Pre-Compile only	true

⌋

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

[SWS_BswM_00269]

Upstream requirements: SRS_ModeMgm_02005, SRS_ModeMgm_02003

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

10.2.58 BswMEcuMGoDownHaltPoll

[ECUC_BswM_00963] Definition of EcucParamConfContainerDef BswMEcuMGoDownHaltPoll ⌈

Container Name	BswMEcuMGoDownHaltPoll
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the action to trigger the EcuM_GoDownHaltPoll from BswM.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMEcuMUserIdRef	1	[ECUC_BswM_00964]

No Included Containers

⌋

[ECUC_BswM_00964] Definition of EcucReferenceDef BswMEcuMUserIdRef ⌈

Parameter Name	BswMEcuMUserIdRef
Parent Container	BswMEcuMGoDownHaltPoll
Upstream mapping defined	–
Description	This is a reference to a EcuM UserId.
Multiplicity	1
Type	Symbolic name reference to EcuMFlexUserConfig
Pre-Compile only	true

⌋

10.2.59 BswMEcuMSelectShutdownTarget

[ECUC_BswM_00961] Definition of EcucParamConfContainerDef BswMEcuMSelectShutdownTarget ⌈

Container Name	BswMEcuMSelectShutdownTarget
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the shutdown target.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEcuMShutdownTarget	1	[ECUC_BswM_00993]
BswMEcuMResetModeRef	0..1	[ECUC_BswM_00994]
BswMEcuMSleepModeRef	0..1	[ECUC_BswM_00962]

No Included Containers

⌋

[ECUC_BswM_00993] Definition of EcucEnumerationParamDef BswMEcuMShutdownTarget ⌈

Parameter Name	BswMEcuMShutdownTarget	
Parent Container	BswMEcuMSelectShutdownTarget	
Upstream mapping defined	–	
Description	This parameter contains the shutdown target that the BswM selects at the EcuM.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_ECUM_SHUTDOWN_TARGET_OFF	Shutdown target off. The configuration parameters BswMEcuMSleepModeRef and BswMEcuMResetModeRef shall not exist.
	BSWM_ECUM_SHUTDOWN_TARGET_RESET	In case the configuration parameter BswMEcuMShutdownTarget is set to BSWM_ECUM_SHUTDOWN_TARGET_RESET the configuration parameter BswMEcuMResetModeRef shall exist and contain a valid reference to a EcuM reset mode.
	BSWM_ECUM_SHUTDOWN_TARGET_SLEEP	In case the configuration parameter BswMEcuMShutdownTarget is set to BSWM_ECUM_SHUTDOWN_TARGET_SLEEP the configuration parameter BswMEcuMSleepModeRef shall exist and contain a valid reference to a EcuM sleep mode.
Pre-Compile only	true	

⌋

[ECUC_BswM_00994] Definition of EcucReferenceDef BswMEcuMResetModeRef

⌈

Parameter Name	BswMEcuMResetModeRef
Parent Container	BswMEcuMSelectShutdownTarget
Upstream mapping defined	–
Description	This is a reference to a reset mode.
Multiplicity	0..1
Type	Symbolic name reference to EcuMResetMode
Pre-Compile only	true

⌋

[ECUC_BswM_00962] Definition of EcucReferenceDef BswMEcuMSleepModeRef

⌈

Parameter Name	BswMEcuMSleepModeRef
Parent Container	BswMEcuMSelectShutdownTarget
Upstream mapping defined	–
Description	This is a reference to a sleep mode.
Multiplicity	0..1
Type	Symbolic name reference to EcuMSleepMode
Pre-Compile only	true

⌋

10.2.60 BswMEcuMStateSwitch

[ECUC_BswM_01045] Definition of EcucParamConfContainerDef BswMEcuMStateSwitch ⌈

Container Name	BswMEcuMStateSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the action to switch a State of the ECU State Manager.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEcuMState	1	[ECUC_BswM_01044]

No Included Containers

⌋

[ECUC_BswM_01044] Definition of EcucEnumerationParamDef BswMEcuMState

⌈

Parameter Name	BswMEcuMState	
Parent Container	BswMEcuMStateSwitch	
Upstream mapping defined	–	
Description	This parameter corresponds to the parameter "State" of the function EcuM_ SetState ().	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_ECUM_STATE_POST_ RUN	Set EcuM state to PostRun
	BSWM_ECUM_STATE_RUN	Set EcuM state to Run
	BSWM_ECUM_STATE_ SHUTDOWN	Set EcuM state to Shutdown
	BSWM_ECUM_STATE_SLEEP	Set EcuM state to Sleep
	BSWM_ECUM_STATE_ STARTUP	Set EcuM state to Startup
Pre-Compile only	true	

⌋

10.2.61 BswMEthIfStartAllPorts

[ECUC_BswM_01088] Definition of EcucParamConfContainerDef BswMEthIf StartAllPorts ⌈

Container Name	BswMEthIfStartAllPorts
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the action to call EthIf_StartAllPorts from BswM and thus to start all affected EthSwtPorts.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

No Included Containers

⌋

[SWS_BswM_00286]

Upstream requirements: SRS_ModeMgm_02007

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

10.2.62 BswMEthIfSwitchPortGroupRequestMode

[ECUC_BswM_01068] Definition of EcucParamConfContainerDef BswMEthIfSwitchPortGroupRequestMode ⌈

Container Name	BswMEthIfSwitchPortGroupRequestMode
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to requesting a mode for the EthIfSwt PortGroup. The EthIf_SwitchPortGroupRequestMode API is called when this action is executed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMEthTrcvMode	1	[ECUC_BswM_01070]
BswMEthIfSwitchPortGroupRef	1	[ECUC_BswM_01069]

No Included Containers

⌋

[ECUC_BswM_01070] Definition of EcucEnumerationParamDef BswMEthTrcvMode ⌈

Parameter Name	BswMEthTrcvMode	
Parent Container	BswMEthIfSwitchPortGroupRequestMode	
Upstream mapping defined	–	
Description	This parameter contains the mode which will be requested.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_ETH_MODE_ACTIVE	enable the port group
	BSWM_ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST	Enable the port group and request to trigger a wake-up on the network.
	BSWM_ETH_MODE_DOWN	disable the port group
Post-Build Value	-	Selectable

⌋

[ECUC_BswM_01069] Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef ⌈

Parameter Name	BswMEthIfSwitchPortGroupRef
Parent Container	BswMEthIfSwitchPortGroupRequestMode
Upstream mapping defined	–
Description	This is a reference to the Ethernet Interface Switch Port Group which will receive the request.

▽

△

Multiplicity	1	
Type	Symbolic name reference to EthIfSwitchPortGroup	
Post-Build Value	-	Selectable

⌋

10.2.63 BswMFrSMAllSlots

[ECUC_BswM_01037] Definition of EcucParamConfContainerDef BswMFrSMAllSlots ⌈

Container Name	BswMFrSMAllSlots
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameter(s) for the action to request an exit from Flexray KeySlotOnlyMode. FrSM_AllSlots is called when this action is executed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMFrSMAllSlotsNetworkHandleRef	1	[ECUC_BswM_01038]

No Included Containers

⌋

[ECUC_BswM_01038] Definition of EcucReferenceDef BswMFrSMAllSlotsNetworkHandleRef ⌈

Parameter Name	BswMFrSMAllSlotsNetworkHandleRef
Parent Container	BswMFrSMAllSlots
Upstream mapping defined	–
Description	This references the FlexRay cluster. The reference corresponds to the parameter "NetworkHandle" of the function FrSM_AllSlots.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.64 BswMIdsMBlockStateChangeRequest

[ECUC_BswM_01089] Definition of EcucParamConfContainerDef BswMIdsMBlockStateChangeRequest

Status: DRAFT

⌈

Container Name	BswMIdsMBlockStateChangeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to switching the Block State of the IdsM. The IdsM_BswM_StateChanged API is called when this action is executed. **Tags:** atp.Status=draft
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMIdsMBlockStateChangeRequestRef	1	[ECUC_BswM_01090]

No Included Containers

⌋

[ECUC_BswM_01090] Definition of EcucReferenceDef BswMIdsMBlockStateChangeRequestRef

Status: DRAFT

⌈

Parameter Name	BswMIdsMBlockStateChangeRequestRef
Parent Container	BswMIdsMBlockStateChangeRequest
Upstream mapping defined	–
Description	This is a reference to the Block State, that shall be set as current Block State of the Ids M. This reference corresponds to the parameter "state" of the function IdsM_BswM_ StateChanged. **Tags:** atp.Status=draft
Multiplicity	1
Type	Symbolic name reference to IdsMBlockState
Pre-Compile only	true

⌋

10.2.65 BswMIdsMTransmissionStateChangeRequest

[ECUC_BswM_01095] Definition of EcucParamConfContainerDef BswMIdsMTransmissionStateChangeRequest ⌈

Container Name	BswMIdsMTransmissionStateChangeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines the action to switch the transmission state of the IdsM. The IdsM_TransmissionSetState API is called when this action is executed.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMIdsMTransmissionRequestedState	1	[ECUC_BswM_01096]

No Included Containers

⌋

[ECUC_BswM_01096] Definition of EcucEnumerationParamDef BswMIdsMTransmissionRequestedState ⌈

Parameter Name	BswMIdsMTransmissionRequestedState	
Parent Container	BswMIdsMTransmissionStateChangeRequest	
Upstream mapping defined	–	
Description	This parameter describes the transmission enable/disable state for IdsM	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_IDSM_TRANSMISSION_STATE_OFF	IdsM transmission state off
	BSWM_IDSM_TRANSMISSION_STATE_ON	IdsM transmission state on
Pre-Compile only	true	

⌋

10.2.66 BswMJ1939DcmStateSwitch

[ECUC_BswM_01032] Definition of EcucParamConfContainerDef BswMJ1939DcmStateSwitch ⌈

Container Name	BswMJ1939DcmStateSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to a switch of the J1939 Diagnostic Communication Managers network state for a J1939 node. J1939Dcm_SetState is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMJ1939DcmRequestedState	1	[ECUC_BswM_01035]
BswMJ1939DcmChannelRef	1	[ECUC_BswM_01033]
BswMJ1939DcmNodeRef	1	[ECUC_BswM_01034]

No Included Containers

⌋

[ECUC_BswM_01035] Definition of EcucEnumerationParamDef BswMJ1939DcmRequestedState ⌈

Parameter Name	BswMJ1939DcmRequestedState	
Parent Container	BswMJ1939DcmStateSwitch	
Upstream mapping defined	–	
Description	This parameter describes the communication state of the J1939 Diagnostic Communication Manager and corresponds to the parameter "newState" of the function J1939Dcm_SetState.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_J1939DCM_STATE_OFFLINE	Set J1939Dcm to offline
	BSWM_J1939DCM_STATE_ONLINE	Set J1939Dcm to online
Pre-Compile only	true	

⌋

[ECUC_BswM_01033] Definition of EcucReferenceDef BswMJ1939DcmChannelRef ⌈

Parameter Name	BswMJ1939DcmChannelRef
Parent Container	BswMJ1939DcmStateSwitch
Upstream mapping defined	–
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 "channel" of the function J1939Dcm_SetState.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

[ECUC_BswM_01034] Definition of EcucReferenceDef BswMJ1939DcmNodeRef ⌈

Parameter Name	BswMJ1939DcmNodeRef
Parent Container	BswMJ1939DcmStateSwitch
Upstream mapping defined	–

▽

△

Description	This reference points to a J1939NmNode and provides access to the unique J1939Nm NodeId.This reference corresponds to the parameter "node" of the function J1939 Dcm_SetState.
Multiplicity	1
Type	Symbolic name reference to J1939NmNode
Pre-Compile only	true

⌋

10.2.67 BswMJ1939RmStateSwitch

[ECUC_BswM_00998] Definition of EcucParamConfContainerDef BswMJ1939Rm StateSwitch ⌈

Container Name	BswMJ1939RmStateSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to a switch of the J1939 Request Managers network state for a J1939 node. J1939Rm_SetState is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMJ1939RmRequestedState	1	[ECUC_BswM_01002]
BswMJ1939RmChannelRef	1	[ECUC_BswM_01000]
BswMJ1939RmNodeRef	1	[ECUC_BswM_01001]

No Included Containers

⌋

[ECUC_BswM_01002] Definition of EcucEnumerationParamDef BswMJ1939Rm RequestedState ⌈

Parameter Name	BswMJ1939RmRequestedState	
Parent Container	BswMJ1939RmStateSwitch	
Upstream mapping defined	–	
Description	This parameter describes the communication state of the J1939 Request Manager and corresponds to the parameter "new state" of the function J1939Rm_SetState.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_J1939RM_STATE_ OFFLINE	Set J1939Rm to offline

▽

△

	BSWM_J1939RM_STATE_ ONLINE	Set J1939Rm to online
Pre-Compile only	true	

⌋

[ECUC_BswM_01000] Definition of EcucReferenceDef BswMJ1939RmChannelRef ⌈

Parameter Name	BswMJ1939RmChannelRef
Parent Container	BswMJ1939RmStateSwitch
Upstream mapping defined	–
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 "channel" of the function J1939Rm_Set State.
Multiplicity	1
Type	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

[ECUC_BswM_01001] Definition of EcucReferenceDef BswMJ1939RmNodeRef ⌈

Parameter Name	BswMJ1939RmNodeRef
Parent Container	BswMJ1939RmStateSwitch
Upstream mapping defined	–
Description	This reference points to a J1939NmNode and provides access to the unique J1939Nm NodeId. This reference corresponds to the parameter "node" of the function J1939Rm_Set State.
Multiplicity	1
Type	Symbolic name reference to J1939NmNode
Pre-Compile only	true

⌋

10.2.68 BswMLinScheduleSwitch

[ECUC_BswM_00827] Definition of EcucParamConfContainerDef BswMLinScheduleSwitch ⌈

Container Name	BswMLinScheduleSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–

▽

△

Description	This container includes all parameters related to a switch of LIN schedule table. Lin SM_ScheduleRequest is called when this action is configured. The configuration for the "network" parameter can be accessed via the reference Lin SMComMNetworkHandleRef contained in the parent container LinSMChannel of the container referenced by BswMLinScheduleRef.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMLinScheduleRef	1	[ECUC_BswM_00842]

No Included Containers

⌋

[ECUC_BswM_00842] Definition of EcucReferenceDef BswMLinScheduleRef ⌈

Parameter Name	BswMLinScheduleRef	
Parent Container	BswMLinScheduleSwitch	
Upstream mapping defined	–	
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	Symbolic name reference to LinSMSchedule	
Post-Build Value	Loadable	Selectable

⌋

10.2.69 BswMNMControl

[ECUC_BswM_00837] Definition of EcucParamConfContainerDef BswMNMControl ⌈

Container Name	BswMNMControl
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMNMAction	1	[ECUC_BswM_00838]
BswMComMNetworkHandleRef	1	[ECUC_BswM_00999]

No Included Containers

⌋

[ECUC_BswM_00838] Definition of EcucEnumerationParamDef BswMNMAction ⌈

Parameter Name	BswMNMAction	
Parent Container	BswMNMControl	
Upstream mapping defined	–	
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	Disable communication on Nm channel
	BSWM_NM_ENABLE	Enable communication on Nm channel
Pre-Compile only	true	

⌋

[ECUC_BswM_00999] Definition of EcucReferenceDef BswMComMNetworkHandleRef ⌈

Parameter Name	BswMComMNetworkHandleRef
Parent Container	BswMNMControl
Upstream mapping defined	–
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	Symbolic name reference to ComMChannel
Pre-Compile only	true

⌋

10.2.70 BswMPduGroupSwitch

[ECUC_BswM_00828] Definition of EcucParamConfContainerDef BswMPduGroupSwitch ⌈

Container Name	BswMPduGroupSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes references to the PDU groups that shall be enabled and disabled.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
BswMPduGroupSwitchReinit	0..1	[ECUC_BswM_00913]
BswMDisabledPduGroupRef	0..*	[ECUC_BswM_00850]
BswMEnabledPduGroupRef	0..*	[ECUC_BswM_00849]

No Included Containers

⌋

[ECUC_BswM_00913] Definition of EcucBooleanParamDef BswMPduGroupSwitchReinit ⌈

Parameter Name	BswMPduGroupSwitchReinit	
Parent Container	BswMPduGroupSwitch	
Upstream mapping defined	–	
Description	This parameter defines if the data of the I-PDU, the shadow buffers of included signal groups, etc. are reinitialized when a PDU Group is started. This parameter corresponds to the parameter "initialize" of the function Com_IpduGroupStart.	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00850] Definition of EcucReferenceDef BswMDisabledPduGroupRef ⌈

Parameter Name	BswMDisabledPduGroupRef	
Parent Container	BswMPduGroupSwitch	
Upstream mapping defined	–	
Description	This is a reference to a PDU Group that should be disabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_IpduGroupStop.	
Multiplicity	0..*	
Type	Symbolic name reference to ComIPduGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00849] Definition of EcucReferenceDef BswMEnabledPduGroupRef ⌈

Parameter Name	BswMEnabledPduGroupRef	
Parent Container	BswMPduGroupSwitch	
Upstream mapping defined	–	
Description	This is a reference to a PDU Group that should be enabled. This reference corresponds to the parameter "IpduGroupId" of the function Com_IpduGroupStart.	
Multiplicity	0..*	
Type	Symbolic name reference to ComIPduGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

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

[SWS_BswM_00273]

Upstream requirements: SRS_ModeMgm_02007, SRS_ModeMgm_02003

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

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

10.2.71 BswMPduRouterControl

[ECUC_BswM_00853] Definition of EcucParamConfContainerDef BswMPduRouterControl ⌈

Container Name	BswMPduRouterControl
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMPduRouterAction	1	[ECUC_BswM_00854]
BswMPduRouterDisableInitBuffer	0..1	[ECUC_BswM_01036]
BswMPduRoutingPathGroupRef	1..*	[ECUC_BswM_00855]

No Included Containers

⌋

[ECUC_BswM_00854] Definition of EcucEnumerationParamDef BswMPduRouterAction ⌈

Parameter Name	BswMPduRouterAction	
Parent Container	BswMPduRouterControl	
Upstream mapping defined	–	
Description	This parameter specifies if the routing of the corresponding PDU should be enabled or disabled.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_PDUR_DISABLE	Disable routing of this PDU
	BSWM_PDUR_ENABLE	Enable routing of this PDU
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_01036] Definition of EcucBooleanParamDef BswMPduRouterDisableInitBuffer ⌈

Parameter Name	BswMPduRouterDisableInitBuffer	
Parent Container	BswMPduRouterControl	
Upstream mapping defined	–	
Description	When BswMPduRouterAction is set to BSWM_PDUR_DISABLE and this parameter is set to true, then the call to PduR_DisableRouting will be invoked with parameter "initialize" set to true, otherwise false.	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00855] Definition of EcucReferenceDef BswMPduRoutingPathGroupRef ⌈

Parameter Name	BswMPduRoutingPathGroupRef
Parent Container	BswMPduRouterControl
Upstream mapping defined	–

▽

△

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	1..*	
Type	Symbolic name reference to PduRRoutingPathGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

10.2.72 BswMRteModeRequest

[ECUC_BswM_01021] Definition of EcucParamConfContainerDef BswMRteModeRequest ⌈

Container Name	BswMRteModeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container defines a mode request that the BswM may send to a SW-C which is acting as a mode-manager. RTE_Write is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMRequestedModeRef	1	[ECUC_BswM_01024]
BswMRteModeRequestPortRef	1	[ECUC_BswM_01023]

No Included Containers

⌋

[ECUC_BswM_01024] Definition of EcucForeignReferenceDef BswMRequestedModeRef ⌈

Parameter Name	BswMRequestedModeRef
Parent Container	BswMRteModeRequest
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is a foreign reference to the Mode Declaration used for the mode request
Multiplicity	1
Type	Foreign reference to MODE-DECLARATION
Pre-Compile only	true

⌋

[ECUC_BswM_01023] Definition of EcucReferenceDef BswMRteModeRequestPortRef ⌈

Parameter Name	BswMRteModeRequestPortRef
Parent Container	BswMRteModeRequest
Upstream mapping defined	–
Description	This is a reference to a BswMRteModeRequestPort.
Multiplicity	1
Type	Reference to BswMRteModeRequestPort
Pre-Compile only	true

⌋

10.2.73 BswMRteSwitch

[ECUC_BswM_00803] Definition of EcucParamConfContainerDef BswMRteSwitch ⌈

Container Name	BswMRteSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMRteSwitchPortRef	1	[ECUC_BswM_00952]
BswMSwitchedMode	1	[ECUC_BswM_00896]

No Included Containers

⌋

[ECUC_BswM_00952] Definition of EcucReferenceDef BswMRteSwitchPortRef ⌈

Parameter Name	BswMRteSwitchPortRef
Parent Container	BswMRteSwitch
Upstream mapping defined	–
Description	This is a reference to the BswMSwitchPort.
Multiplicity	1
Type	Reference to BswMSwitchPort
Pre-Compile only	true

⌋

[ECUC_BswM_00896] Definition of EcucForeignReferenceDef BswMSwitchedMode ⌈

Parameter Name	BswMSwitchedMode
Parent Container	BswMRteSwitch
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
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
Pre-Compile only	true

⌋

10.2.74 BswMSchMSwitch

[ECUC_BswM_00899] Definition of EcucParamConfContainerDef BswMSchMSwitch ⌈

Container Name	BswMSchMSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSchMSwitchedMode	1	[ECUC_BswM_00901]
BswMSchMSwitchPortRef	1	[ECUC_BswM_01080]

No Included Containers

⌋

[ECUC_BswM_00901] Definition of EcucForeignReferenceDef BswMSchMSwitchedMode ⌈

Parameter Name	BswMSchMSwitchedMode
Parent Container	BswMSchMSwitch
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
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

▽

△

Pre-Compile only	true

⌋

[ECUC_BswM_01080] Definition of EcucReferenceDef BswMSchMSwitchPortRef ⌈

Parameter Name	BswMSchMSwitchPortRef
Parent Container	BswMSchMSwitch
Upstream mapping defined	–
Description	This is a reference to the BswMSwitchPort.
Multiplicity	1
Type	Reference to BswMSwitchPort
Pre-Compile only	true

⌋

10.2.75 BswMSdClientServiceModeRequest

[ECUC_BswM_00974] Definition of EcucParamConfContainerDef BswMSdClient ServiceModeRequest ⌈

Container Name	BswMSdClientServiceModeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to the selection of an client service of Sd. Sd_ClientServiceSetState is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdClientServiceState	1	[ECUC_BswM_01016]
BswMSdClientMethodsRef	1	[ECUC_BswM_01009]

No Included Containers

⌋

[ECUC_BswM_01016] Definition of EcucEnumerationParamDef BswMSdClient ServiceState ⌈

Parameter Name	BswMSdClientServiceState
Parent Container	BswMSdClientServiceModeRequest
Upstream mapping defined	–

▽

△

Description	This parameter specifies if the corresponding client service shall be released or requested.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_SD_CLIENT_SERVICE_RELEASED	Client service shall be released
	BSWM_SD_CLIENT_SERVICE_REQUESTED	Client service shall be requested
Pre-Compile only	true	

⌋

For parameter table [ECUC_BswM_01009] BswMSdClientMethodsRef, see definition below container BswMSdClientServiceCurrentState.

10.2.76 BswMSdConsumedEventGroupModeRequest

[ECUC_BswM_01004] Definition of EcucParamConfContainerDef BswMSdConsumedEventGroupModeRequest ⌈

Container Name	BswMSdConsumedEventGroupModeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to the selection of a consumed Event Group of Sd. Sd_ConsumedEventGroupSetState is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdConsumedEventGroupState	1	[ECUC_BswM_01017]
BswMSdConsumedEventGroupRef	1	[ECUC_BswM_01010]

No Included Containers

⌋

[ECUC_BswM_01017] Definition of EcucEnumerationParamDef BswMSdConsumedEventGroupState ⌈

Parameter Name	BswMSdConsumedEventGroupState
Parent Container	BswMSdConsumedEventGroupModeRequest
Upstream mapping defined	–

▽

△

Description	This parameter specifies if the corresponding consumed event group shall be released or requested.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_SD_CONSUMED_ EVENTGROUP_RELEASED	Event group shall be released.
	BSWM_SD_CONSUMED_ EVENTGROUP_REQUESTED	Event group shall be requested.
Pre-Compile only	true	

⌋

For parameter table [ECUC_BswM_01010] BswMSdConsumedEventGroupRef, see definition below container BswMSdConsumedEventGroupCurrentState.

10.2.77 BswMSdServerServiceModeRequest

[ECUC_BswM_01005] Definition of EcucParamConfContainerDef BswMSdServerServiceModeRequest ⌈

Container Name	BswMSdServerServiceModeRequest
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters related to the selection of a server service of Sd. Sd_ServerServiceSetState is called when this action is configured.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSdServerServiceState	1	[ECUC_BswM_01015]
BswMSdServerMethodsRef	1	[ECUC_BswM_01007]

No Included Containers

⌋

[ECUC_BswM_01015] Definition of EcucEnumerationParamDef BswMSdServerServiceState ⌈

Parameter Name	BswMSdServerServiceState
Parent Container	BswMSdServerServiceModeRequest
Upstream mapping defined	–
Description	This parameter specifies if the corresponding server service shall be down or available.
Multiplicity	1

▽

△

Type	EcucEnumerationParamDef	
Range	BSWM_SD_SERVER_SERVICE_AVAILABLE	Server service shall be available.
	BSWM_SD_SERVER_SERVICE_DOWN	Server service shall be down.
Pre-Compile only	true	

⌋

[ECUC_BswM_01007] Definition of EcucReferenceDef BswMSdServerMethodsRef ⌈

Parameter Name	BswMSdServerMethodsRef
Parent Container	BswMSdServerServiceModeRequest
Upstream mapping defined	–
Description	This is a reference to a server service in the Sd module.
Multiplicity	1
Type	Symbolic name reference to SdServerService
Pre-Compile only	true

⌋

10.2.78 BswMSdServiceGroupSwitch

[ECUC_BswM_01077] Definition of EcucParamConfContainerDef BswMSdServiceGroupSwitch ⌈

Container Name	BswMSdServiceGroupSwitch
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes references to the SdServiceGroups that shall be enabled and disabled.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMDisabledSdServiceGroupRef	0..*	[ECUC_BswM_01078]
BswMEnabledSdServiceGroupRef	0..*	[ECUC_BswM_01079]

No Included Containers

⌋

[ECUC_BswM_01078] Definition of EcucReferenceDef BswMDisabledSdService GroupRef ⌈

Parameter Name	BswMDisabledSdServiceGroupRef	
Parent Container	BswMSdServiceGroupSwitch	
Upstream mapping defined	–	
Description	This is a reference to a SdServiceGroup that should be disabled. This reference corresponds to the parameter "ServiceGroupId" of the function Sd_ServiceGroupStop.	
Multiplicity	0..*	
Type	Symbolic name reference to SdServiceGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_01079] Definition of EcucReferenceDef BswMEnabledSdService GroupRef ⌈

Parameter Name	BswMEnabledSdServiceGroupRef	
Parent Container	BswMSdServiceGroupSwitch	
Upstream mapping defined	–	
Description	This is a reference to a SdServiceGroup that should be enabled. This reference corresponds to the parameter "ServiceGroupId" of the function Sd_ServiceGroupStart.	
Multiplicity	0..*	
Type	Symbolic name reference to SdServiceGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[SWS_BswM_CONSTR_00005] ⌈The BswM shall reject configurations where a `BswMSdServiceGroupSwitch` container has a `BswMDisabledSdServiceGroupRef` and a `BswMEnabledSdServiceGroupRef` which reference the same SdServiceGroup.⌋

[SWS_BswM_00280]

Upstream requirements: SRS_ModeMgm_02008

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

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

10.2.79 BswMSetFirewallState

[ECUC_BswM_01094] Definition of EcucParamConfContainerDef BswMSetFirewallState

Status: DRAFT

⌈

Container Name	BswMSetFirewallState
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes parameters for switching the state of the Firewall. The BswM will call Fw_SetFirewallState() when this action is executed. **Tags:** atp.Status=draft
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMFirewallStateRef	1	[ECUC_BswM_00145]

No Included Containers

⌋

[ECUC_BswM_00145] Definition of EcucReferenceDef BswMFirewallStateRef

Status: DRAFT

⌈

Parameter Name	BswMFirewallStateRef	
Parent Container	BswMSetFirewallState	
Upstream mapping defined	–	
Description	Reference to the Firewall State that is activated. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	Symbolic name reference to FirewallState	
Post-Build Value	Loadable	Selectable

⌋

10.2.80 BswMSwitchIPduMode

[ECUC_BswM_00958] Definition of EcucParamConfContainerDef BswMSwitchIPduMode ⌈

Container Name	BswMSwitchIPduMode
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMSwitchIPduModeValue	1	[ECUC_BswM_00960]
BswMSwitchIPduModeRef	1	[ECUC_BswM_00959]

No Included Containers

⌋

[ECUC_BswM_00960] Definition of EcucBooleanParamDef BswMSwitchIPduModeValue ⌈

Parameter Name	BswMSwitchIPduModeValue	
Parent Container	BswMSwitchIPduMode	
Upstream mapping defined	–	
Description	This parameter defines which transmission mode shall be selected during this call. This parameter corresponds to the parameter "Mode" of the function Com_SwitchIpduTxMode.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00959] Definition of EcucReferenceDef BswMSwitchIPduModeRef ⌈

Parameter Name	BswMSwitchIPduModeRef	
Parent Container	BswMSwitchIPduMode	
Upstream mapping defined	–	
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	Symbolic name reference to ComIPdu	
Post-Build Value	Loadable	Selectable

⌋

10.2.81 BswMTimerControl

[ECUC_BswM_01059] Definition of EcucParamConfContainerDef BswMTimerControl ⌈

Container Name	BswMTimerControl
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all parameters for the action to start or to stop a timer.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMTimerAction	1	[ECUC_BswM_01060]
BswMTimerValue	0..1	[ECUC_BswM_01061]
BswMTimerRef	1	[ECUC_BswM_01062]

No Included Containers

⌋

[ECUC_BswM_01060] Definition of EcucEnumerationParamDef BswMTimerAction ⌈

Parameter Name	BswMTimerAction	
Parent Container	BswMTimerControl	
Upstream mapping defined	–	
Description	Specify the action for the timer. The timer can be started or stopped.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_TIMER_START	Start the timer
	BSWM_TIMER_STOP	Stop the timer
Default value	BSWM_TIMER_START	
Pre-Compile only	true	

⌋

[ECUC_BswM_01061] Definition of EcucFloatParamDef BswMTimerValue ⌈

Parameter Name	BswMTimerValue	
Parent Container	BswMTimerControl	
Upstream mapping defined	–	
Description	Specify the timer value (in seconds) that is used when the timer is started.	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	]0 .. INF[	

▽

△

Default value	–
Pre-Compile only	true

⌋

[ECUC_BswM_01062] Definition of EcucReferenceDef BswMTimerRef ⌈

Parameter Name	BswMTimerRef
Parent Container	BswMTimerControl
Upstream mapping defined	–
Description	Specify the Timer for which the timer action shall be executed.
Multiplicity	1
Type	Reference to BswMTimer
Pre-Compile only	true

⌋

10.2.82 BswMTriggerIPduSend

[ECUC_BswM_00906] Definition of EcucParamConfContainerDef BswMTriggerIPduSend ⌈

Container Name	BswMTriggerIPduSend
Parent Container	BswMAvailableActions
Upstream mapping defined	–
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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMTriggeredIPduRef	1..*	[ECUC_BswM_00907]

No Included Containers

⌋

[ECUC_BswM_00907] Definition of EcucReferenceDef BswMTriggeredIPduRef ⌈

Parameter Name	BswMTriggeredIPduRef
Parent Container	BswMTriggerIPduSend
Upstream mapping defined	–

▽

△

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_Trigger IPDUSend.	
Multiplicity	1..*	
Type	Symbolic name reference to ComIPdu	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

10.2.83 BswMUserCallout

[ECUC_BswM_00834] Definition of EcucParamConfContainerDef BswMUserCallout ⌈

Container Name	BswMUserCallout
Parent Container	BswMAvailableActions
Upstream mapping defined	–
Description	This container includes all details needed for a user defined function call.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMUserCalloutFunction	1	[ECUC_BswM_00843]

No Included Containers

⌋

[ECUC_BswM_00843] Definition of EcucStringParamDef BswMUserCalloutFunction ⌈

Parameter Name	BswMUserCalloutFunction
Parent Container	BswMUserCallout
Upstream mapping defined	–
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 NvM_ReadAll() Action to call NvM_WriteAll()
Multiplicity	1
Type	EcucStringParamDef
Default value	–
Regular Expression	–

▽

△

Pre-Compile only	true

⌋

10.2.84 BswMActionList

[ECUC_BswM_00809] Definition of EcucParamConfContainerDef BswMAction List ⌈

Container Name	BswMActionList	
Parent Container	BswMModeControl	
Upstream mapping defined	–	
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.	
Multiplicity	0..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMActionListExecution	1	[ECUC_BswM_00894]
BswMActionListPriority	0..1	[ECUC_BswM_01076]

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

⌋

[ECUC_BswM_00894] Definition of EcucEnumerationParamDef BswMActionList Execution ⌈

Parameter Name	BswMActionListExecution	
Parent Container	BswMActionList	
Upstream mapping defined	–	
Description	This parameter controls if the corresponding action list shall be executed every time the rule is evaluated or only when the result of the evaluation changes. This parameter does not have an effect when this action list is executed within another action list.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BSWM_CONDITION	Action list shall be executed every time the rule is evaluated.
	BSWM_TRIGGER	Action list shall be executed every time the result of the evaluation changes.
Pre-Compile only	true	

⌋

[ECUC_BswM_01076] Definition of EcucIntegerParamDef BswMActionListPriority ⌈

Parameter Name	BswMActionListPriority	
Parent Container	BswMActionList	
Upstream mapping defined	–	
Description	This controls the order of execution, in the case when multiple action lists are executed during a single mode control cycle. Highest priority action list will be executed first. Zero (0) is lowest priority, and 255 is highest priority.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	0	
Pre-Compile only	true	

⌋

10.2.85 BswMActionListItem

[ECUC_BswM_00823] Definition of EcucParamConfContainerDef BswMActionListItem ⌈

Container Name	BswMActionListItem	
Parent Container	BswMActionList	
Upstream mapping defined	–	
Description	This container defines an item in an action list.	
Multiplicity	1..*	
Post-Build Multiplicity	Loadable	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMAbortOnFail	1	[ECUC_BswM_00902]
BswMActionListItemIndex	1	[ECUC_BswM_00824]
BswMReportFailRuntimeErrorId	0..1	[ECUC_BswM_01050]
BswMActionListItemRef	1	[ECUC_BswM_00825]

No Included Containers

⌋

[ECUC_BswM_00902] Definition of EcucBooleanParamDef BswMAbortOnFail ⌈

Parameter Name	BswMAbortOnFail
Parent Container	BswMActionListItem
Upstream mapping defined	–

▽

△

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 and as such has no effect when the associated BswMActionListItemRef references a BswMActionList or BswMRule, as these elements do not provide a return value.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00824] Definition of EcucIntegerParamDef BswMActionListItem Index ⌈

Parameter Name	BswMActionListItemIndex	
Parent Container	BswMActionListItem	
Upstream mapping defined	–	
Description	This parameter defines the index of the action in the action list. It is used define in which order the actions shall be performed.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_01050] Definition of EcucIntegerParamDef BswMReportFailRuntimeErrorId ⌈

Parameter Name	BswMReportFailRuntimeErrorId	
Parent Container	BswMActionListItem	
Upstream mapping defined	–	
Description	If this parameter is configured, and this specific action returns E_NOT_OK, the BswM will report a Det Runtime Error. The ErrorId reported in the Runtime Error is given by the value configured in this parameter.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	128 .. 255	
Default value	128	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_BswM_00825] Definition of EcucChoiceReferenceDef BswMActionListItemRef ⌈

Parameter Name	BswMActionListItemRef	
Parent Container	BswMActionListItem	
Upstream mapping defined	–	
Description	The action item can either be an atomic action or a reference to another action list or rule.	
Multiplicity	1	
Type	Choice reference to [BswMAction, BswMActionList, BswMRule]	
Post-Build Value	Loadable	Selectable

⌋

10.2.86 BswMRteModeRequestPort

[ECUC_BswM_01022] Definition of EcucParamConfContainerDef BswMRteModeRequestPort ⌈

Container Name	BswMRteModeRequestPort
Parent Container	BswMModeControl
Upstream mapping defined	–
Description	This container defines a mode request port which the BswM may utilize to send a mode request to a SW-C which is acting as a mode-manager. If this container is referenced by a BswMRteModeRequest, the BswM shall create a corresponding PPort in its service description.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMRteModeRequestPortInterfaceRef	0..1	[ECUC_BswM_01025]
BswMRteModeRequestVariableDataPrototypeSRRef	0..1	[ECUC_BswM_01057]

No Included Containers

⌋

[ECUC_BswM_01025] Definition of EcucInstanceReferenceDef BswMRteModeRequestPortInterfaceRef ⌈

Parameter Name	BswMRteModeRequestPortInterfaceRef
Parent Container	BswMRteModeRequestPort
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is an instance reference to the variable data prototype used for the mode request. Dependency: BswMRteModeRequestVariableDataPrototypeSRRef == NULL
Multiplicity	0..1

▽

△

Type	Instance reference to VARIABLE-DATA-PROTOTYPE context: SW-COMPONENT-PROTOTYPE* PORT-PROTOTYPE
Pre-Compile only	true

⌋

[ECUC_BswM_01057] Definition of EcucForeignReferenceDef BswMRteModeRequestVariableDataPrototypeSRRef ⌈

Parameter Name	BswMRteModeRequestVariableDataPrototypeSRRef
Parent Container	BswMRteModeRequestPort
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	This is a foreign reference to a VariableDataPrototype used for the mode request. Dependency: BswMRteModeRequestPortInterfaceRef == NULL
Multiplicity	0..1
Type	Foreign reference to VARIABLE-DATA-PROTOTYPE
Pre-Compile only	true

⌋

10.2.87 BswMSwitchPort

[ECUC_BswM_00950] Definition of EcucParamConfContainerDef BswMSwitchPort ⌈

Container Name	BswMSwitchPort
Parent Container	BswMModeControl
Upstream mapping defined	–
Description	Represents an output mode-switch port to be generated by the BswM. If BswMModeSwitchInterfaceRef is configured then a PPortPrototype is generated in the SWCD. If BswMSchMModeDeclarationGroupRef is configured then a ModeDeclarationGroupPrototype is generated in the ProvidedModeGroups of the BSWMD. If both BswMModeSwitchInterfaceRef and BswMSchMModeDeclarationGroupRef are configured then an SwcBswSynchronizedModeGroupPrototype is also generated in the BSWMD (see Chapter 6.11 of the BSW Module Description Template SWS and EXP ModemanagementGuide)..
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMModeSwitchInterfaceRef	0..1	[ECUC_BswM_00951]
BswMSchMModeDeclarationGroupRef	0..1	[ECUC_BswM_01031]

No Included Containers

⌋

[ECUC_BswM_00951] Definition of EcucForeignReferenceDef BswMModeSwitch InterfaceRef ⌈

Parameter Name	BswMModeSwitchInterfaceRef
Parent Container	BswMSwitchPort
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	Reference to the ModeSwitchInterface from which the BswM will generate a PPort Prototype.
Multiplicity	0..1
Type	Foreign reference to MODE-SWITCH-INTERFACE
Pre-Compile only	true

⌋

[ECUC_BswM_01031] Definition of EcucForeignReferenceDef BswMSchMMode DeclarationGroupRef ⌈

Parameter Name	BswMSchMModeDeclarationGroupRef
Parent Container	BswMSwitchPort
Upstream mapping defined	[6, CP_TPS_SoftwareComponentTemplate]
Description	Reference to the ModeDeclarationGroup from which the BswM will generate a Mode DeclarationGroupPrototype.
Multiplicity	0..1
Type	Foreign reference to MODE-DECLARATION-GROUP
Pre-Compile only	true

⌋

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

10.2.88 BswMGeneral

[ECUC_BswM_00800] Definition of EcucParamConfContainerDef BswMGeneral ⌈

Container Name	BswMGeneral
Parent Container	BswM
Upstream mapping defined	–
Description	General configuration parameters of the Basic SW Mode Manager.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMCanSMEnabled	1	[ECUC_BswM_00938]
BswMComMEnabled	1	[ECUC_BswM_00939]
BswMDcmEnabled	1	[ECUC_BswM_00940]
BswMDevErrorDetect	1	[ECUC_BswM_00811]
BswMEcuMEnabled	1	[ECUC_BswM_00941]
BswMEthIfEnabled	1	[ECUC_BswM_01072]
BswMEthSMEnabled	1	[ECUC_BswM_00942]
BswMFrSMEnabled	1	[ECUC_BswM_00943]
BswMGenericRequestEnabled	1	[ECUC_BswM_00949]
BswMJ1939DcmEnabled	1	[ECUC_BswM_00987]
BswMJ1939NmEnabled	1	[ECUC_BswM_00965]
BswMLinSMEnabled	1	[ECUC_BswM_00944]
BswMLinTPEnabled	1	[ECUC_BswM_00945]
BswMMainFunctionPeriod	0..1	[ECUC_BswM_00813]
BswMMkaEnabled	1	[ECUC_BswM_01100]
BswMNmEnabled	1	[ECUC_BswM_01071]
BswMNvMEnabled	1	[ECUC_BswM_00946]
BswMSdControlEnabled	1	[ECUC_BswM_01093]
BswMSdEnabled	1	[ECUC_BswM_01047]
BswMSoAdEnabled	1	[ECUC_BswM_01097]

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

⌋

[ECUC_BswM_00938] Definition of EcucBooleanParamDef BswMCanSMEnabled

⌈

Parameter Name	BswMCanSMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable CanSM module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00939] Definition of EcucBooleanParamDef BswMComMEnabled ⌈

Parameter Name	BswMComMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable ComM module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00940] Definition of EcucBooleanParamDef BswMDcmEnabled ⌈

Parameter Name	BswMDcmEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable Dcm module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00811] Definition of EcucBooleanParamDef BswMDevErrorDetect ⌈

Parameter Name	BswMDevErrorDetect
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00941] Definition of EcucBooleanParamDef BswMEcuMEnabled ⌈

Parameter Name	BswMEcuMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable EcuM module related BswM API: true: Enabled false: Disabled
Multiplicity	1

▽

△

Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_01072] Definition of EcucBooleanParamDef BswMEthIfEnabled ⌈

Parameter Name	BswMEthIfEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable EthIf module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00942] Definition of EcucBooleanParamDef BswMEthSMEnabled ⌈

Parameter Name	BswMEthSMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable EthSM module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00943] Definition of EcucBooleanParamDef BswMFrSMEnabled ⌈

Parameter Name	BswMFrSMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable FrSM module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00949] Definition of EcucBooleanParamDef BswMGenericRequestEnabled ⌈

Parameter Name	BswMGenericRequestEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable Generic Request related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00987] Definition of EcucBooleanParamDef BswMJ1939DcmEnabled ⌈

Parameter Name	BswMJ1939DcmEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	Enable/disable J1939Dcm module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00965] Definition of EcucBooleanParamDef BswMJ1939NmEnabled ⌈

Parameter Name	BswMJ1939NmEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	Enable/disable J1939Nm module related BswM API. true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00944] Definition of EcucBooleanParamDef BswMLinSMEnabled ⌈

Parameter Name	BswMLinSMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable LinSM module related BswM API: true: Enabled false: Disabled
Multiplicity	1

▽

△

Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00945] Definition of EcucBooleanParamDef BswMLinTPEnabled ⌈

Parameter Name	BswMLinTPEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable LinTP module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00813] Definition of EcucFloatParamDef BswMMainFunctionPeriod ⌈

Parameter Name	BswMMainFunctionPeriod
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	The cycle time of the periodic main function of BswM. Defined in seconds .
Multiplicity	0..1
Type	EcucFloatParamDef
Range	]0 .. INF[
Default value	–
Pre-Compile only	true

⌋

[ECUC_BswM_01100] Definition of EcucBooleanParamDef BswMMkaEnabled ⌈

Parameter Name	BswMMkaEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable MKA module related BswM API: • true: Enabled • false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_01071] Definition of EcucBooleanParamDef BswMNmEnabled ⌈

Parameter Name	BswMNmEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable Nm module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_00946] Definition of EcucBooleanParamDef BswMNvMEnabled ⌈

Parameter Name	BswMNvMEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable NvM module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_01093] Definition of EcucBooleanParamDef BswMSdControlEnabled ⌈

Parameter Name	BswMSdControlEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable ServiceDiscovery Control for Applications: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_01047] Definition of EcucBooleanParamDef BswMSdEnabled ⌈

Parameter Name	BswMSdEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable Sd module related BswM API. true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef

▽

△

Default value	false
Pre-Compile only	true

⌋

[ECUC_BswM_01097] Definition of EcucBooleanParamDef BswMSoAdEnabled ⌈

Parameter Name	BswMSoAdEnabled
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	enable/disable SoAd module related BswM API: true: Enabled false: Disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[SWS_BswM_00279]

Upstream requirements: SRS_BSW_00301

⌈The availability of private header files (e.g. BswM_ComM.h) between the BswM and other basic software modules depend on the configuration parameters (BswM<BSWModule>Enabled). If the configuration parameter is set to false, then no private header shall be provided by BswM for that BSWModule.⌋

10.2.89 BswMUserIncludeFiles

[ECUC_BswM_00954] Definition of EcucParamConfContainerDef BswMUserIncludeFiles ⌈

Container Name	BswMUserIncludeFiles
Parent Container	BswMGeneral
Upstream mapping defined	–
Description	Collection of header file names which shall be included by the BswM.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
BswMUserIncludeFile	1..*	[ECUC_BswM_00955]

No Included Containers

⌋

[ECUC_BswM_00955] Definition of EcucStringParamDef BswMUserIncludeFile ⌈

Parameter Name	BswMUserIncludeFile
Parent Container	BswMUserIncludeFiles
Upstream mapping defined	–
Description	Header file name which shall be included by the BswM. The value of this parameter shall be used as h-char-sequence or q-char-sequence according to ISO C99. The parameter value shall not represent a path.
Multiplicity	1..*
Type	EcucStringParamDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.3 Published Information

For details refer to [1] Chapter 10.3 *"Published Information"*.

A Not applicable requirements

[SWS_BswM_NA_09999]

Upstream requirements: SRS_BSW_00170, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00339, SRS_BSW_00409

⌈These requirements are not applicable to this specification.⌋

B Change history of AUTOSAR traceable items

B.1 Traceable item history of this document according to AUTOSAR Release R23-11

B.1.1 Added Specification Items in R23-11

Number	Heading
[SWS_BSWM_91006]	Definition of ImplementationDataType BswM_ClientModeRequestEnum
[SWS_BSWM_91007]	Definition of Port ClientServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91008]	Definition of ImplementationDataType BswM_OfferModeEnum
[SWS_BSWM_91009]	Definition of SenderReceiverInterface ServerServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91010]	Definition of SenderReceiverInterface ClientServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91011]	Definition of Port ServerServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91012]	Definition of Port ServerServiceSubscriptionState_{RuleName}_{SwcName_ PortName} provided by module BswM
[SWS_BSWM_91013]	Definition of ModeSwitchInterface ServerServiceSubscriptionState
[SWS_BSWM_91014]	Definition of ModeDeclarationGroup MDG_ServerServiceSubscriptionState
[SWS_BSWM_91015]	Definition of Port ClientServiceSubscriptionState_{RuleName}_{SwcName_ PortName} provided by module BswM
[SWS_BSWM_91016]	Definition of ModeSwitchInterface ClientServiceSubscriptionState
[SWS_BSWM_91018]	Definition of ModeDeclarationGroup MDG_ClientServiceSubscriptionState
[SWS_BswM_00287]	Event Subscription Rule Creation
[SWS_BswM_00288]	Service Request Rule Creation
[SWS_BswM_00289]	Client Subscription SenderReceiver Interface
[SWS_BswM_00290]	Client Subscription RPort
[SWS_BswM_00291]	Input for Event Subscription Rule
[SWS_BswM_00292]	Input for Service Request Rule
[SWS_BswM_00293]	Content of Event Subscription rule
[SWS_BswM_00294]	Content of Client Service request rule
[SWS_BswM_00295]	Client Service Request TRUE ActionList
[SWS_BswM_00296]	Client Service Request FALSE ActionList
[SWS_BswM_00297]	Order of ActionList Execution
[SWS_BswM_00298]	Event Subscription TRUE ActionList
[SWS_BswM_00299]	Event Subscription FALSE ActionList
[SWS_BswM_00300]	Mode Switch Interface ClientServiceSubscriptionState
[SWS_BswM_00301]	Service Availability Notification Rule creation

▽

△

Number	Heading
[SWS_BswM_00302]	BswMModeRequestSource BswMSdConsumedEventGroup CurrentState
[SWS_BswM_00303]	BswMModeRequestSource BswMSdClientServiceCurrentState
[SWS_BswM_00304]	Client Event Subscription notification rule
[SWS_BswM_00305]	Client Subscription Notification PPort
[SWS_BswM_00306]	Client Service availability TRUE action list
[SWS_BswM_00307]	Client Service availability FALSE action list
[SWS_BswM_00308]	Sender Receiver Interface ServerServiceRequest
[SWS_BswM_00309]	Service Offer Rule Creation
[SWS_BswM_00310]	Service Offer Port Creation
[SWS_BswM_00311]	Service Offer Rule Input
[SWS_BswM_00312]	Service Offer rule
[SWS_BswM_00313]	Service Offer TRUE ActionList
[SWS_BswM_00314]	Service Offer FALSE ActionList
[SWS_BswM_00315]	MSIF ServerServiceSubscriptionState
[SWS_BswM_00316]	Server Event Subscription Notification Rule creation
[SWS_BswM_00317]	Server Subscription Notification PPort
[SWS_BswM_00318]	BswMModeRequestSource BswMSdEventHandlerCurrentState
[SWS_BswM_00319]	Client Service availability notification rule content

Table B.1: Added Specification Items in R23-11

B.1.2 Changed Specification Items in R23-11

Number	Heading
[SWS_BswM_00008]	Definition of optional interfaces in module BswM

Table B.2: Changed Specification Items in R23-11

B.1.3 Deleted Specification Items in R23-11

none

B.1.4 Added Constraints in R23-11

Number	Heading
[SWS_BswM_CONSTR_00007]	Unique port for offering a service instance

Table B.3: Added Constraints in R23-11

B.1.5 Changed Constraints in R23-11

none

B.1.6 Deleted Constraints in R23-11

none

B.1.7 Added Specification Items in R24-11

Number	Heading
[ECUC_BswM_01089]	Definition of EcucParamConfContainerDef BswMIdsMBlockStateChange Request
[ECUC_BswM_01090]	Definition of EcucReferenceDef BswMIdsMBlockStateChangeRequestRef
[ECUC_BswM_01095]	Definition of EcucParamConfContainerDef BswMIdsMTransmissionState ChangeRequest
[ECUC_BswM_01096]	Definition of EcucEnumerationParamDef BswMIdsMTransmissionRequested State
[SWS_BswM_00320]	Partition Local Action Lists

Table B.4: Added Specification Items in R24-11

B.1.8 Changed Specification Items in R24-11

Number	Heading
[ECUC_BswM_00826]	Definition of EcucChoiceContainerDef BswMAvailableActions
[ECUC_BswM_01025]	Definition of EcucInstanceReferenceDef BswMRteModeRequestPort InterfaceRef

▽

△

Number	Heading
[ECUC_BswM_01036]	Definition of EcucBooleanParamDef BswMPduRouterDisableInitBuffer
[SWS_BSWM_91006]	Definition of ImplementationDataType BswM_ClientModeRequestEnum
[SWS_BSWM_91007]	Definition of Port ClientServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91008]	Definition of ImplementationDataType BswM_OfferModeEnum
[SWS_BSWM_91009]	Definition of SenderReceiverInterface ServerServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91010]	Definition of SenderReceiverInterface ClientServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91011]	Definition of Port ServerServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91012]	Definition of Port ServerServiceSubscriptionState_{RuleName}_{SwcName_PortName} provided by module BswM
[SWS_BSWM_91013]	Definition of ModeSwitchInterface ServerServiceSubscriptionState
[SWS_BSWM_91014]	Definition of ModeDeclarationGroup MDG_ServerServiceSubscriptionState
[SWS_BSWM_91015]	Definition of Port ClientServiceSubscriptionState_{RuleName}_{SwcName_PortName} provided by module BswM
[SWS_BSWM_91016]	Definition of ModeSwitchInterface ClientServiceSubscriptionState
[SWS_BSWM_91018]	Definition of ModeDeclarationGroup MDG_ClientServiceSubscriptionState
[SWS_BswM_00001]	Definition of imported datatypes of module BswM
[SWS_BswM_00008]	Definition of optional interfaces requested by module BswM
[SWS_BswM_00230]	Definiton of development errors in module BswM
[SWS_BswM_00273]	
[SWS_BswM_00274]	
[SWS_BswM_00275]	
[SWS_BswM_00276]	
[SWS_BswM_00277]	
[SWS_BswM_00278]	
[SWS_BswM_00280]	
[SWS_BswM_00281]	
[SWS_BswM_00282]	
[SWS_BswM_00284]	
[SWS_BswM_00285]	
[SWS_BswM_00286]	
[SWS_BswM_00287]	Event Subscription Rule Creation
[SWS_BswM_00288]	Service Request Rule Creation
[SWS_BswM_00289]	Client Subscription SenderReceiver Interface
[SWS_BswM_00290]	Client Subscription RPort
[SWS_BswM_00291]	Input for Event Subscription Rule

▽

△

Number	Heading
[SWS_BswM_00292]	Input for Service Request Rule
[SWS_BswM_00293]	Content of Event Subscription rule
[SWS_BswM_00294]	Content of Client Service request rule
[SWS_BswM_00295]	Client Service Request TRUE ActionList
[SWS_BswM_00296]	Client Service Request FALSE ActionList
[SWS_BswM_00297]	Order of ActionList Execution
[SWS_BswM_00298]	Event Subscription TRUE ActionList
[SWS_BswM_00299]	Event Subscription FALSE ActionList
[SWS_BswM_00300]	Mode Switch Interface ClientServiceSubscriptionState
[SWS_BswM_00301]	Service Availability Notification Rule creation
[SWS_BswM_00302]	BswMModeRequestSource BswMSdConsumedEventGroup CurrentState
[SWS_BswM_00303]	BswMModeRequestSource BswMSdClientServiceCurrentState
[SWS_BswM_00304]	Client Event Subscription notification rule
[SWS_BswM_00305]	Client Subscription Notification PPort
[SWS_BswM_00306]	Client Service availability TRUE action list
[SWS_BswM_00307]	Client Service availability FALSE action list
[SWS_BswM_00308]	Sender Receiver Interface ServerServiceRequest
[SWS_BswM_00309]	Service Offer Rule Creation
[SWS_BswM_00310]	Service Offer Port Creation
[SWS_BswM_00311]	Service Offer Rule Input
[SWS_BswM_00312]	Service Offer rule
[SWS_BswM_00313]	Service Offer TRUE ActionList
[SWS_BswM_00314]	Service Offer FALSE ActionList
[SWS_BswM_00315]	MSIF ServerServiceSubscriptionState
[SWS_BswM_00316]	Server Event Subscription Notification Rule creation
[SWS_BswM_00317]	Server Subscription Notification PPort
[SWS_BswM_00318]	BswMModeRequestSource BswMSdEventHandlerCurrentState
[SWS_BswM_00319]	Client Service availability notification rule content
[SWS_BswM_91002]	Definition of API function BswM_Nm_StateChangeNotification
[SWS_BswM_91004]	Definition of API function BswM_EcuM_RequestedState
[SWS_BswM_91005]	Definition of API function BswM_SoAd_SoConModeChg

Table B.5: Changed Specification Items in R24-11

B.1.9 Deleted Specification Items in R24-11

none

B.1.10 Added Constraints in R24-11

none

B.1.11 Changed Constraints in R24-11

Number	Heading
[SWS_BswM_CONSTR_00001]	
[SWS_BswM_CONSTR_00002]	
[SWS_BswM_CONSTR_00007]	Unique port for offering a service instance

Table B.6: Changed Constraints in R24-11

B.1.12 Deleted Constraints in R24-11

none

B.1.13 Added Specification Items in R25-11

Number	Heading
[ECUC_BswM_01097]	Definition of EcucBooleanParamDef BswMSoAdEnabled
[SWS_BswM_00321]	Development Errors for Invalid Network Handle
[SWS_BswM_00322]	Development Errors for Invalid Handle
[SWS_BswM_00323]	Development Errors for Out of Range
[SWS_BswM_00324]	Development Errors for other Parameters

Table B.7: Added Specification Items in R25-11

B.1.14 Changed Specification Items in R25-11

Number	Heading
[ECUC_BswM_00145]	Definition of EcucReferenceDef BswMFirewallStateRef
[ECUC_BswM_00800]	Definition of EcucParamConfContainerDef BswMGeneral
[ECUC_BswM_00801]	Definition of EcucParamConfContainerDef BswMArbitration
[ECUC_BswM_00802]	Definition of EcucParamConfContainerDef BswMModeControl
[ECUC_BswM_00803]	Definition of EcucParamConfContainerDef BswMRteSwitch
[ECUC_BswM_00805]	Definition of EcucParamConfContainerDef BswMModeRequestPort
[ECUC_BswM_00806]	Definition of EcucParamConfContainerDef BswMRule
[ECUC_BswM_00807]	Definition of EcucParamConfContainerDef BswMModeCondition
[ECUC_BswM_00808]	Definition of EcucParamConfContainerDef BswMLogicalExpression
[ECUC_BswM_00809]	Definition of EcucParamConfContainerDef BswMActionList
[ECUC_BswM_00810]	Definition of EcucParamConfContainerDef BswMAction
[ECUC_BswM_00811]	Definition of EcucBooleanParamDef BswMDevErrorDetect
[ECUC_BswM_00812]	Definition of EcucBooleanParamDef BswMVersionInfoApi
[ECUC_BswM_00813]	Definition of EcucFloatParamDef BswMMainFunctionPeriod
[ECUC_BswM_00814]	Definition of EcucEnumerationParamDef BswMLogicalOperator
[ECUC_BswM_00815]	Definition of EcucEnumerationParamDef BswMConditionType
[ECUC_BswM_00816]	Definition of EcucChoiceContainerDef BswMConditionValue
[ECUC_BswM_00817]	Definition of EcucReferenceDef BswMRuleTrueActionList
[ECUC_BswM_00818]	Definition of EcucReferenceDef BswMRuleFalseActionList
[ECUC_BswM_00819]	Definition of EcucReferenceDef BswMRuleExpressionRef

▽

△

Number	Heading
[ECUC_BswM_00820]	Definition of EcucChoiceReferenceDef BswMArgumentRef
[ECUC_BswM_00821]	Definition of EcucChoiceReferenceDef BswMConditionMode
[ECUC_BswM_00822]	Definition of EcucEnumerationParamDef BswMRequestProcessing
[ECUC_BswM_00823]	Definition of EcucParamConfContainerDef BswMActionListItem
[ECUC_BswM_00824]	Definition of EcucIntegerParamDef BswMActionListItemIndex
[ECUC_BswM_00825]	Definition of EcucChoiceReferenceDef BswMActionListItemRef
[ECUC_BswM_00826]	Definition of EcucChoiceContainerDef BswMAvailableActions
[ECUC_BswM_00827]	Definition of EcucParamConfContainerDef BswMLinScheduleSwitch
[ECUC_BswM_00828]	Definition of EcucParamConfContainerDef BswMPduGroupSwitch
[ECUC_BswM_00830]	Definition of EcucParamConfContainerDef BswMDeadlineMonitoringControl
[ECUC_BswM_00831]	Definition of EcucParamConfContainerDef BswMComMModeSwitch
[ECUC_BswM_00834]	Definition of EcucParamConfContainerDef BswMUserCallout
[ECUC_BswM_00837]	Definition of EcucParamConfContainerDef BswMNMControl
[ECUC_BswM_00838]	Definition of EcucEnumerationParamDef BswMNMAction
[ECUC_BswM_00840]	Definition of EcucEnumerationParamDef BswMComMRequestedMode
[ECUC_BswM_00841]	Definition of EcucReferenceDef BswMComMUserRef
[ECUC_BswM_00842]	Definition of EcucReferenceDef BswMLinScheduleRef
[ECUC_BswM_00843]	Definition of EcucStringParamDef BswMUserCalloutFunction
[ECUC_BswM_00849]	Definition of EcucReferenceDef BswMEnabledPduGroupRef
[ECUC_BswM_00850]	Definition of EcucReferenceDef BswMDisabledPduGroupRef
[ECUC_BswM_00851]	Definition of EcucReferenceDef BswMEnabledDMPduGroupRef
[ECUC_BswM_00852]	Definition of EcucReferenceDef BswMDisabledDMPduGroupRef

▽

△

Number	Heading
[ECUC_BswM_00853]	Definition of EcucParamConfContainerDef BswMPduRouterControl
[ECUC_BswM_00854]	Definition of EcucEnumerationParamDef BswMPduRouterAction
[ECUC_BswM_00855]	Definition of EcucReferenceDef BswMPduRoutingPathGroupRef
[ECUC_BswM_00856]	Definition of EcucChoiceContainerDef BswMModeRequestSource
[ECUC_BswM_00857]	Definition of EcucParamConfContainerDef BswMCanSMIndication
[ECUC_BswM_00858]	Definition of EcucParamConfContainerDef BswMFrSMIndication
[ECUC_BswM_00859]	Definition of EcucParamConfContainerDef BswMLinSMIndication
[ECUC_BswM_00860]	Definition of EcucParamConfContainerDef BswMEthSMIndication
[ECUC_BswM_00861]	Definition of EcucParamConfContainerDef BswMGenericRequest
[ECUC_BswM_00862]	Definition of EcucParamConfContainerDef BswMSwcModeRequest
[ECUC_BswM_00863]	Definition of EcucParamConfContainerDef BswMDcmComModeRequest
[ECUC_BswM_00864]	Definition of EcucForeignReferenceDef BswMModeValueRef
[ECUC_BswM_00866]	Definition of EcucStringParamDef BswMBswRequestedMode
[ECUC_BswM_00868]	Definition of EcucParamConfContainerDef BswMModeDeclaration
[ECUC_BswM_00869]	Definition of EcucParamConfContainerDef BswMBswMode
[ECUC_BswM_00870]	Definition of EcucReferenceDef BswMCanSMChannelRef
[ECUC_BswM_00871]	Definition of EcucReferenceDef BswMLinSMChannelRef
[ECUC_BswM_00872]	Definition of EcucReferenceDef BswMFrSMChannelRef
[ECUC_BswM_00873]	Definition of EcucReferenceDef BswMEthSMChannelRef
[ECUC_BswM_00874]	Definition of EcucIntegerParamDef BswMModeRequesterId
[ECUC_BswM_00880]	Definition of EcucParamConfContainerDef BswMComMIndication
[ECUC_BswM_00883]	Definition of EcucReferenceDef BswMComMChannelRef

▽

△

Number	Heading
[ECUC_BswM_00885]	Definition of EcucParamConfContainerDef BswMLinScheduleIndication
[ECUC_BswM_00888]	Definition of EcucEnumerationParamDef BswMRuleInitState
[ECUC_BswM_00890]	Definition of EcucParamConfContainerDef BswMNvMRequest
[ECUC_BswM_00891]	Definition of EcucReferenceDef BswMNvMBlockRef
[ECUC_BswM_00892]	Definition of EcucParamConfContainerDef BswMSwcModeNotification
[ECUC_BswM_00893]	Definition of EcucForeignReferenceDef BswMSwcModeNotificationMode DeclarationGroupPrototypeRef
[ECUC_BswM_00894]	Definition of EcucEnumerationParamDef BswMActionListExecution
[ECUC_BswM_00895]	Definition of EcucParamConfContainerDef BswMConfig
[ECUC_BswM_00896]	Definition of EcucForeignReferenceDef BswMSwitchedMode
[ECUC_BswM_00899]	Definition of EcucParamConfContainerDef BswMSchMSwitch
[ECUC_BswM_00901]	Definition of EcucForeignReferenceDef BswMSchMSwitchedMode
[ECUC_BswM_00902]	Definition of EcucBooleanParamDef BswMAbortOnFail
[ECUC_BswM_00904]	Definition of EcucParamConfContainerDef BswMEcuMWakeupSource
[ECUC_BswM_00905]	Definition of EcucReferenceDef BswMEcuMWakeupSrcRef
[ECUC_BswM_00906]	Definition of EcucParamConfContainerDef BswMTriggerIPduSend
[ECUC_BswM_00907]	Definition of EcucReferenceDef BswMTriggeredIPduRef
[ECUC_BswM_00908]	Definition of EcucParamConfContainerDef BswMComMModeLimitation
[ECUC_BswM_00909]	Definition of EcucParamConfContainerDef BswMComMAllowCom
[ECUC_BswM_00910]	Definition of EcucBooleanParamDef BswMComMLimitMode
[ECUC_BswM_00911]	Definition of EcucReferenceDef BswMComMLimitChannelRef
[ECUC_BswM_00912]	Definition of EcucReferenceDef BswMComMAllowChannelRef
[ECUC_BswM_00913]	Definition of EcucBooleanParamDef BswMPduGroupSwitchReinit

▽

△

Number	Heading
[ECUC_BswM_00914]	Definition of EcucParamConfContainerDef BswMLinTpModeRequest
[ECUC_BswM_00915]	Definition of EcucReferenceDef BswMLinTpChannelRef
[ECUC_BswM_00918]	Definition of EcucBooleanParamDef BswMComAllowed
[ECUC_BswM_00922]	Definition of EcucParamConfContainerDef BswMComMPncRequest
[ECUC_BswM_00924]	Definition of EcucReferenceDef BswMComMPncRef
[ECUC_BswM_00925]	Definition of EcucParamConfContainerDef BswMDcmApplicationUpdated Indication
[ECUC_BswM_00926]	Definition of EcucParamConfContainerDef BswMBswModeNotification
[ECUC_BswM_00927]	Definition of EcucForeignReferenceDef BswMBswModeDeclarationGroup PrototypeRef
[ECUC_BswM_00928]	Definition of EcucParamConfContainerDef BswMModeInitValue
[ECUC_BswM_00932]	Definition of EcucStringParamDef BswMBswModeInitValue
[ECUC_BswM_00935]	Definition of EcucBooleanParamDef BswMNestedExecutionOnly
[ECUC_BswM_00936]	Definition of EcucParamConfContainerDef BswMDataTypeMappingSets
[ECUC_BswM_00937]	Definition of EcucForeignReferenceDef BswMDataTypeMappingSetRef
[ECUC_BswM_00938]	Definition of EcucBooleanParamDef BswMCanSMEnabled
[ECUC_BswM_00939]	Definition of EcucBooleanParamDef BswMComMEnabled
[ECUC_BswM_00940]	Definition of EcucBooleanParamDef BswMDcmEnabled
[ECUC_BswM_00941]	Definition of EcucBooleanParamDef BswMEcuMEnabled
[ECUC_BswM_00942]	Definition of EcucBooleanParamDef BswMEthSMEnabled
[ECUC_BswM_00943]	Definition of EcucBooleanParamDef BswMFrSMEnabled
[ECUC_BswM_00944]	Definition of EcucBooleanParamDef BswMLinSMEnabled
[ECUC_BswM_00945]	Definition of EcucBooleanParamDef BswMLinTPEnabled
[ECUC_BswM_00946]	Definition of EcucBooleanParamDef BswMNvMEnabled

▽

△

Number	Heading
[ECUC_BswM_00949]	Definition of EcucBooleanParamDef BswMGenericRequestEnabled
[ECUC_BswM_00950]	Definition of EcucParamConfContainerDef BswMSwitchPort
[ECUC_BswM_00951]	Definition of EcucForeignReferenceDef BswMModeSwitchInterfaceRef
[ECUC_BswM_00952]	Definition of EcucReferenceDef BswMRteSwitchPortRef
[ECUC_BswM_00954]	Definition of EcucParamConfContainerDef BswMUserIncludeFiles
[ECUC_BswM_00955]	Definition of EcucStringParamDef BswMUserIncludeFile
[ECUC_BswM_00956]	Definition of EcucParamConfContainerDef BswMNvMJobModeIndication
[ECUC_BswM_00957]	Definition of EcucEnumerationParamDef BswMNvmService
[ECUC_BswM_00958]	Definition of EcucParamConfContainerDef BswMSwitchIPduMode
[ECUC_BswM_00959]	Definition of EcucReferenceDef BswMSwitchIPduModeRef
[ECUC_BswM_00960]	Definition of EcucBooleanParamDef BswMSwitchIPduModeValue
[ECUC_BswM_00961]	Definition of EcucParamConfContainerDef BswMEcuMSelectShutdown Target
[ECUC_BswM_00962]	Definition of EcucReferenceDef BswMEcuMSleepModeRef
[ECUC_BswM_00963]	Definition of EcucParamConfContainerDef BswMEcuMGoDownHaltPoll
[ECUC_BswM_00964]	Definition of EcucReferenceDef BswMEcuMUserIdRef
[ECUC_BswM_00965]	Definition of EcucBooleanParamDef BswMJ1939NmEnabled
[ECUC_BswM_00966]	Definition of EcucParamConfContainerDef BswMJ1939NmIndication
[ECUC_BswM_00967]	Definition of EcucReferenceDef BswMJ1939NmChannelRef
[ECUC_BswM_00969]	Definition of EcucReferenceDef BswMDcmComMChannelRef
[ECUC_BswM_00970]	Definition of EcucParamConfContainerDef BswMCoreHaltMode
[ECUC_BswM_00971]	Definition of EcucReferenceDef BswMTargetCoreRef
[ECUC_BswM_00972]	Definition of EcucStringParamDef BswMCoreHaltActivationState

▽

△

Number	Heading
[ECUC_BswM_00974]	Definition of EcucParamConfContainerDef BswMSdClientServiceMode Request
[ECUC_BswM_00984]	Definition of EcucReferenceDef BswMPartitionRef
[ECUC_BswM_00985]	Definition of EcucParamConfContainerDef BswMJ1939DcmBroadcastStatus
[ECUC_BswM_00987]	Definition of EcucBooleanParamDef BswMJ1939DcmEnabled
[ECUC_BswM_00988]	Definition of EcucReferenceDef BswMJ1939DcmChannelRef
[ECUC_BswM_00989]	Definition of EcucParamConfContainerDef BswMPartitionRestarted
[ECUC_BswM_00990]	Definition of EcucParamConfContainerDef BswMModeSwitchErrorEvent
[ECUC_BswM_00993]	Definition of EcucEnumerationParamDef BswMEcuMShutdownTarget
[ECUC_BswM_00994]	Definition of EcucReferenceDef BswMEcuMResetModeRef
[ECUC_BswM_00997]	Definition of EcucReferenceDef BswMJ1939NmNodeRef
[ECUC_BswM_00998]	Definition of EcucParamConfContainerDef BswMJ1939RmStateSwitch
[ECUC_BswM_00999]	Definition of EcucReferenceDef BswMComMNetworkHandleRef
[ECUC_BswM_01000]	Definition of EcucReferenceDef BswMJ1939RmChannelRef
[ECUC_BswM_01001]	Definition of EcucReferenceDef BswMJ1939RmNodeRef
[ECUC_BswM_01002]	Definition of EcucEnumerationParamDef BswMJ1939RmRequestedState
[ECUC_BswM_01004]	Definition of EcucParamConfContainerDef BswMSdConsumedEventGroup ModeRequest
[ECUC_BswM_01005]	Definition of EcucParamConfContainerDef BswMSdServerServiceMode Request
[ECUC_BswM_01007]	Definition of EcucReferenceDef BswMSdServerMethodsRef
[ECUC_BswM_01008]	Definition of EcucReferenceDef BswMSdEventHandlerRef
[ECUC_BswM_01009]	Definition of EcucReferenceDef BswMSdClientMethodsRef
[ECUC_BswM_01010]	Definition of EcucReferenceDef BswMSdConsumedEventGroupRef
[ECUC_BswM_01011]	Definition of EcucParamConfContainerDef BswMSdClientServiceCurrent State

▽

△

Number	Heading
[ECUC_BswM_01012]	Definition of EcucParamConfContainerDef BswMSdConsumedEventGroupCurrentState
[ECUC_BswM_01013]	Definition of EcucParamConfContainerDef BswMSdEventHandlerCurrentState
[ECUC_BswM_01014]	Definition of EcucParamConfContainerDef BswMComMInitiateReset
[ECUC_BswM_01015]	Definition of EcucEnumerationParamDef BswMSdServerServiceState
[ECUC_BswM_01016]	Definition of EcucEnumerationParamDef BswMSdClientServiceState
[ECUC_BswM_01017]	Definition of EcucEnumerationParamDef BswMSdConsumedEventGroupState
[ECUC_BswM_01021]	Definition of EcucParamConfContainerDef BswMRteModeRequest
[ECUC_BswM_01022]	Definition of EcucParamConfContainerDef BswMRteModeRequestPort
[ECUC_BswM_01023]	Definition of EcucReferenceDef BswMRteModeRequestPortRef
[ECUC_BswM_01024]	Definition of EcucForeignReferenceDef BswMRequestedModeRef
[ECUC_BswM_01025]	Definition of EcucInstanceReferenceDef BswMRteModeRequestPortInterfaceRef
[ECUC_BswM_01028]	Definition of EcucReferenceDef BswMLinSMChannelRef
[ECUC_BswM_01030]	Definition of EcucReferenceDef BswMRteSwitchPortRef
[ECUC_BswM_01031]	Definition of EcucForeignReferenceDef BswMSchMModeDeclarationGroupRef
[ECUC_BswM_01032]	Definition of EcucParamConfContainerDef BswMJ1939DcmStateSwitch
[ECUC_BswM_01033]	Definition of EcucReferenceDef BswMJ1939DcmChannelRef
[ECUC_BswM_01034]	Definition of EcucReferenceDef BswMJ1939DcmNodeRef
[ECUC_BswM_01035]	Definition of EcucEnumerationParamDef BswMJ1939DcmRequestedState
[ECUC_BswM_01036]	Definition of EcucBooleanParamDef BswMPduRouterDisableInitBuffer
[ECUC_BswM_01037]	Definition of EcucParamConfContainerDef BswMFrSMAllSlots
[ECUC_BswM_01038]	Definition of EcucReferenceDef BswMFrSMAllSlotsNetworkHandleRef
[ECUC_BswM_01039]	Definition of EcucParamConfContainerDef BswMCompuScaleModeValue

▽

△

Number	Heading
[ECUC_BswM_01040]	Definition of EcucForeignReferenceDef BswMCompuMethodRef
[ECUC_BswM_01041]	Definition of EcucStringParamDef BswMCompuConstText
[ECUC_BswM_01044]	Definition of EcucEnumerationParamDef BswMEcuMState
[ECUC_BswM_01045]	Definition of EcucParamConfContainerDef BswMEcuMStateSwitch
[ECUC_BswM_01046]	Definition of EcucForeignReferenceDef BswMSwcModeRequestVariable DataPrototypeRef
[ECUC_BswM_01047]	Definition of EcucBooleanParamDef BswMSdEnabled
[ECUC_BswM_01049]	Definition of EcucReferenceDef BswMNmChannelRef
[ECUC_BswM_01050]	Definition of EcucIntegerParamDef BswMReportFailRuntimeErrorId
[ECUC_BswM_01052]	Definition of EcucParamConfContainerDef BswMEventRequestPort
[ECUC_BswM_01053]	Definition of EcucChoiceContainerDef BswMEventRequestSource
[ECUC_BswM_01054]	Definition of EcucParamConfContainerDef BswMClearEventRequest
[ECUC_BswM_01055]	Definition of EcucReferenceDef BswMClearEventRequestPortRef
[ECUC_BswM_01056]	Definition of EcucEnumerationParamDef BswMEventRequestProcessing
[ECUC_BswM_01057]	Definition of EcucForeignReferenceDef BswMRteModeRequestVariableData PrototypeSRRef
[ECUC_BswM_01058]	Definition of EcucParamConfContainerDef BswMTimer
[ECUC_BswM_01059]	Definition of EcucParamConfContainerDef BswMTimerControl
[ECUC_BswM_01060]	Definition of EcucEnumerationParamDef BswMTimerAction
[ECUC_BswM_01061]	Definition of EcucFloatParamDef BswMTimerValue
[ECUC_BswM_01062]	Definition of EcucReferenceDef BswMTimerRef
[ECUC_BswM_01063]	Definition of EcucModuleDef BswM
[ECUC_BswM_01064]	Definition of EcucParamConfContainerDef BswMEcuMDriverInitListBswM
[ECUC_BswM_01065]	Definition of EcucReferenceDef BswMEcuMDriverInitListBswMRef

▽

△

Number	Heading
[ECUC_BswM_01066]	Definition of EcucParamConfContainerDef BswMEthIfPortGroupLinkStateChg
[ECUC_BswM_01067]	Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef
[ECUC_BswM_01068]	Definition of EcucParamConfContainerDef BswMEthIfSwitchPortGroupRequestMode
[ECUC_BswM_01069]	Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef
[ECUC_BswM_01070]	Definition of EcucEnumerationParamDef BswMEthTrcvMode
[ECUC_BswM_01071]	Definition of EcucBooleanParamDef BswMNmEnabled
[ECUC_BswM_01072]	Definition of EcucBooleanParamDef BswMEthIfEnabled
[ECUC_BswM_01073]	Definition of EcucParamConfContainerDef BswMRteStart
[ECUC_BswM_01074]	Definition of EcucParamConfContainerDef BswMRteStop
[ECUC_BswM_01075]	Definition of EcucParamConfContainerDef BswMNmCarWakeUpIndication
[ECUC_BswM_01076]	Definition of EcucIntegerParamDef BswMActionListPriority
[ECUC_BswM_01077]	Definition of EcucParamConfContainerDef BswMSdServiceGroupSwitch
[ECUC_BswM_01078]	Definition of EcucReferenceDef BswMDisabledSdServiceGroupRef
[ECUC_BswM_01079]	Definition of EcucReferenceDef BswMEnabledSdServiceGroupRef
[ECUC_BswM_01080]	Definition of EcucReferenceDef BswMSchMSwitchPortRef
[ECUC_BswM_01081]	Definition of EcucParamConfContainerDef BswMNmStateChangeNotification
[ECUC_BswM_01082]	Definition of EcucReferenceDef BswMNmChannelRef
[ECUC_BswM_01083]	Definition of EcucParamConfContainerDef BswMSwitchAckNotification
[ECUC_BswM_01084]	Definition of EcucReferenceDef BswMSwitchPortRef
[ECUC_BswM_01085]	Definition of EcucParamConfContainerDef BswMEcuMIndication
[ECUC_BswM_01086]	Definition of EcucParamConfContainerDef BswMEcuMRUNRequestIndication
[ECUC_BswM_01087]	Definition of EcucEnumerationParamDef BswMEcuMRUNRequestProtocolPort

▽

△

Number	Heading
[ECUC_BswM_01088]	Definition of EcucParamConfContainerDef BswMEthIfStartAllPorts
[ECUC_BswM_01089]	Definition of EcucParamConfContainerDef BswMIdsMBlockStateChange Request
[ECUC_BswM_01090]	Definition of EcucReferenceDef BswMIdsMBlockStateChangeRequestRef
[ECUC_BswM_01091]	Definition of EcucParamConfContainerDef BswMSoAdSoConModeChg
[ECUC_BswM_01092]	Definition of EcucReferenceDef BswMSoAdSocketConnectionRef
[ECUC_BswM_01093]	Definition of EcucBooleanParamDef BswMSdControlEnabled
[ECUC_BswM_01094]	Definition of EcucParamConfContainerDef BswMSetFirewallState
[ECUC_BswM_01095]	Definition of EcucParamConfContainerDef BswMIdsMTransmissionState ChangeRequest
[ECUC_BswM_01096]	Definition of EcucEnumerationParamDef BswMIdsMTransmissionRequested State
[SWS_BSWM_91006]	Definition of ImplementationDataType BswM_ClientModeRequestType
[SWS_BSWM_91007]	Definition of Port ClientServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91008]	Definition of ImplementationDataType BswM_OfferModeType
[SWS_BSWM_91009]	Definition of SenderReceiverInterface ServerServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91010]	Definition of SenderReceiverInterface ClientServiceRequest_{Arb Name}_{PortName}
[SWS_BSWM_91011]	Definition of Port ServerServiceRequest_{ArbName}_{PortName} required by module BswM
[SWS_BSWM_91012]	Definition of Port ServerServiceSubscriptionState_{RuleName}_{SwcName_ PortName} provided by module BswM
[SWS_BSWM_91013]	Definition of ModeSwitchInterface ServerServiceSubscriptionState
[SWS_BSWM_91014]	Definition of ModeDeclarationGroup MDG_ServerServiceSubscriptionState
[SWS_BSWM_91015]	Definition of Port ClientServiceSubscriptionState_{RuleName}_{SwcName_ PortName} provided by module BswM
[SWS_BSWM_91016]	Definition of ModeSwitchInterface ClientServiceSubscriptionState
[SWS_BSWM_91018]	Definition of ModeDeclarationGroup MDG_ClientServiceSubscriptionState
[SWS_BswM_00001]	Definition of imported datatypes of module BswM
[SWS_BswM_00008]	Definition of optional interfaces requested by module BswM
[SWS_BswM_00009]	
[SWS_BswM_00010]	
[SWS_BswM_00011]	
[SWS_BswM_00012]	

▽

△

Number	Heading
[SWS_BswM_00013]	
[SWS_BswM_00014]	
[SWS_BswM_00015]	
[SWS_BswM_00016]	
[SWS_BswM_00017]	
[SWS_BswM_00018]	
[SWS_BswM_00019]	
[SWS_BswM_00023]	
[SWS_BswM_00035]	
[SWS_BswM_00037]	
[SWS_BswM_00038]	
[SWS_BswM_00039]	
[SWS_BswM_00040]	
[SWS_BswM_00046]	Definition of API function BswM_RequestMode
[SWS_BswM_00047]	Definition of API function BswM_ComM_CurrentMode
[SWS_BswM_00048]	Definition of API function BswM_Dcm_CommunicationMode_CurrentState
[SWS_BswM_00049]	Definition of API function BswM_CanSM_CurrentState
[SWS_BswM_00050]	Definition of API function BswM_EthSM_CurrentState
[SWS_BswM_00051]	Definition of API function BswM_FrSM_CurrentState
[SWS_BswM_00052]	Definition of API function BswM_LinSM_CurrentState
[SWS_BswM_00053]	Definition of scheduled function BswM_MainFunction
[SWS_BswM_00054]	
[SWS_BswM_00055]	
[SWS_BswM_00058]	Definition of API function BswM_LinSM_CurrentSchedule
[SWS_BswM_00059]	
[SWS_BswM_00060]	
[SWS_BswM_00061]	
[SWS_BswM_00062]	
[SWS_BswM_00064]	
[SWS_BswM_00066]	The BswM shall act according to what is stated in this table when a rule is evaluated for the first time after initialization. Note: The "true" and "false" action lists are optional for each rule.
[SWS_BswM_00067]	
[SWS_BswM_00068]	
[SWS_BswM_00069]	
[SWS_BswM_00075]	
[SWS_BswM_00104]	Definition of API function BswM_NvM_CurrentBlockMode
[SWS_BswM_00115]	
[SWS_BswM_00116]	

▽

△

Number	Heading
[SWS_BswM_00117]	
[SWS_BswM_00131]	Definition of API function BswM_EcuM_CurrentWakeup
[SWS_BswM_00147]	
[SWS_BswM_00148]	Definition of API function BswM_ComM_CurrentPNCMode
[SWS_BswM_00152]	Definition of API function BswM_NvM_CurrentJobMode
[SWS_BswM_00156]	Definition of API function BswM_LinTp_RequestMode
[SWS_BswM_00158]	Definition of API function BswM_Dcm_ApplicationUpdated
[SWS_BswM_00165]	Definition of API function BswM_J1939DcmBroadcastStatus
[SWS_BswM_00189]	
[SWS_BswM_00193]	Definition of API function BswM_BswMPartitionRestarted
[SWS_BswM_00194]	Definition of API function BswM_J1939Nm_StateChangeNotification
[SWS_BswM_00200]	Definition of Port modeNotificationPort_{ArbName}_{ModeName} required by module BswM
[SWS_BswM_00201]	Definition of Port modeRequestPort_{ArbName}_{ReqName} required by module BswM
[SWS_BswM_00202]	Definition of Port modeSwitchPort_{ModConName}_{SwitchName} provided by module BswM
[SWS_BswM_00203]	
[SWS_BswM_00204]	Definition of API function BswM_Sd_ClientServiceCurrentState
[SWS_BswM_00207]	Definition of API function BswM_Sd_ConsumedEventGroupCurrentState
[SWS_BswM_00210]	Definition of API function BswM_Sd_EventHandlerCurrentState
[SWS_BswM_00213]	Definition of datatype BswM_ConfigType
[SWS_BswM_00214]	Definition of datatype BswM_ModeType
[SWS_BswM_00216]	Definition of datatype BswM_UserType
[SWS_BswM_00217]	Definition of API function BswM_ComM_InitiateReset
[SWS_BswM_00220]	
[SWS_BswM_00223]	
[SWS_BswM_00230]	Definition of development errors in module BswM
[SWS_BswM_00235]	Definition of API function BswM_Nm_CarWakeUpIndication
[SWS_BswM_00236]	
[SWS_BswM_00241]	
[SWS_BswM_00242]	
[SWS_BswM_00243]	
[SWS_BswM_00244]	
[SWS_BswM_00245]	
[SWS_BswM_00246]	
[SWS_BswM_00247]	
[SWS_BswM_00248]	
[SWS_BswM_00249]	

▽

△

Number	Heading
[SWS_BswM_00250]	
[SWS_BswM_00251]	
[SWS_BswM_00252]	
[SWS_BswM_00253]	
[SWS_BswM_00254]	
[SWS_BswM_00255]	
[SWS_BswM_00256]	Configuration constraint for mode request comparisons and event request checking
[SWS_BswM_00257]	
[SWS_BswM_00258]	
[SWS_BswM_00259]	
[SWS_BswM_00260]	
[SWS_BswM_00261]	
[SWS_BswM_00262]	
[SWS_BswM_00263]	
[SWS_BswM_00264]	
[SWS_BswM_00265]	
[SWS_BswM_00269]	
[SWS_BswM_00271]	
[SWS_BswM_00272]	
[SWS_BswM_00273]	
[SWS_BswM_00274]	
[SWS_BswM_00275]	
[SWS_BswM_00276]	
[SWS_BswM_00277]	
[SWS_BswM_00278]	
[SWS_BswM_00280]	
[SWS_BswM_00281]	
[SWS_BswM_00282]	
[SWS_BswM_00283]	API to Request Port Mappings: This table describes which request port shall be triggered when an API is called. In addition, it describes which configuration parameter(s) correspond to which API parameter(s)
[SWS_BswM_00284]	
[SWS_BswM_00285]	
[SWS_BswM_00286]	
[SWS_BswM_00287]	Event Subscription Rule Creation
[SWS_BswM_00288]	Service Request Rule Creation
[SWS_BswM_00289]	Client Subscription SenderReceiver Interface
[SWS_BswM_00290]	Client Subscription RPort

▽

△

Number	Heading
[SWS_BswM_00291]	Input for Event Subscription Rule
[SWS_BswM_00292]	Input for Service Request Rule
[SWS_BswM_00293]	Content of Event Subscription rule
[SWS_BswM_00294]	Content of Client Service request rule
[SWS_BswM_00295]	Client Service Request TRUE ActionList
[SWS_BswM_00296]	Client Service Request FALSE ActionList
[SWS_BswM_00297]	Order of ActionList Execution
[SWS_BswM_00298]	Event Subscription TRUE ActionList
[SWS_BswM_00299]	Event Subscription FALSE ActionList
[SWS_BswM_00300]	Mode Switch Interface ClientServiceSubscriptionState
[SWS_BswM_00301]	Service Availability Notification Rule creation
[SWS_BswM_00302]	BswMModeRequestSource BswMSdConsumedEventGroup CurrentState
[SWS_BswM_00303]	BswMModeRequestSource BswMSdClientServiceCurrentState
[SWS_BswM_00304]	Client Event Subscription notification rule
[SWS_BswM_00305]	Client Subscription Notification PPort
[SWS_BswM_00306]	Client Service availability TRUE action list
[SWS_BswM_00307]	Client Service availability FALSE action list
[SWS_BswM_00308]	Sender Receiver Interface ServerServiceRequest
[SWS_BswM_00309]	Service Offer Rule Creation
[SWS_BswM_00310]	Service Offer Port Creation
[SWS_BswM_00311]	Service Offer Rule Input
[SWS_BswM_00312]	Service Offer rule
[SWS_BswM_00313]	Service Offer TRUE ActionList
[SWS_BswM_00314]	Service Offer FALSE ActionList
[SWS_BswM_00315]	MSIF ServerServiceSubscriptionState
[SWS_BswM_00316]	Server Event Subscription Notification Rule creation
[SWS_BswM_00317]	Server Subscription Notification PPort
[SWS_BswM_00318]	BswMModeRequestSource BswMSdEventHandlerCurrentState
[SWS_BswM_00319]	Client Service availability notification rule content
[SWS_BswM_00320]	Partition Local Action Lists
[SWS_BswM_91001]	Definition of API function BswM_EthIf_PortGroupLinkStateChg
[SWS_BswM_91002]	Definition of API function BswM_Nm_StateChangeNotification
[SWS_BswM_91003]	Definition of API function BswM_EcuM_CurrentState
[SWS_BswM_91004]	Definition of API function BswM_EcuM_RequestedState
[SWS_BswM_91005]	Definition of callback function BswM_SoAd_SoConModeChg

Table B.8: Changed Specification Items in R25-11

B.1.15 Deleted Specification Items in R25-11

Number	Heading
[ECUC_BswM_00886]	Definition of EcucReferenceDef BswMLinScheduleRef
[SWS_BswM_00045]	
[SWS_BswM_00077]	
[SWS_BswM_00078]	
[SWS_BswM_00079]	
[SWS_BswM_00080]	
[SWS_BswM_00081]	
[SWS_BswM_00082]	
[SWS_BswM_00083]	
[SWS_BswM_00084]	
[SWS_BswM_00086]	
[SWS_BswM_00089]	
[SWS_BswM_00090]	
[SWS_BswM_00091]	
[SWS_BswM_00093]	
[SWS_BswM_00095]	
[SWS_BswM_00097]	
[SWS_BswM_00099]	
[SWS_BswM_00101]	
[SWS_BswM_00103]	
[SWS_BswM_00109]	
[SWS_BswM_00110]	
[SWS_BswM_00112]	
[SWS_BswM_00113]	
[SWS_BswM_00132]	
[SWS_BswM_00133]	
[SWS_BswM_00149]	
[SWS_BswM_00150]	
[SWS_BswM_00153]	
[SWS_BswM_00154]	
[SWS_BswM_00159]	
[SWS_BswM_00205]	
[SWS_BswM_00206]	
[SWS_BswM_00208]	
[SWS_BswM_00209]	

▽

△

Number	Heading
[SWS_BswM_00211]	
[SWS_BswM_00212]	
[SWS_BswM_00227]	
[SWS_BswM_00228]	
[SWS_BswM_00229]	
[SWS_BswM_00266]	
[SWS_BswM_00267]	
[SWS_BswM_00268]	

Table B.9: Deleted Specification Items in R25-11

B.1.16 Added Constraints in R25-11

none

B.1.17 Changed Constraints in R25-11

Number	Heading
[SWS_BswM_CONSTR_00007]	Unique port for offering a service instance

Table B.10: Changed Constraints in R25-11

B.1.18 Deleted Constraints in R25-11

none

B.1.19 Added Specification Items in R26-11

Number	Heading
[ECUC_BswM_01098]	Definition of EcucParamConfContainerDef BswMMkaControlledPortStatus Indication
[ECUC_BswM_01099]	Definition of EcucReferenceDef BswMMkaChannelRef
[ECUC_BswM_01100]	Definition of EcucBooleanParamDef BswMMkaEnabled

▽

△

Number	Heading
[SWS_BswM_00325]	Arbitration Independence
[SWS_BswM_00326]	Deferred Rule Arbitration for Separate BswM Instances
[SWS_BswM_91019]	Definition of API function BswM_Mka_ControlledPortStatus

Table B.11: Added Specification Items in R26-11

B.1.20 Changed Specification Items in R26-11

Number	Heading
[ECUC_BswM_00145]	Definition of EcucReferenceDef BswMFirewallStateRef
[ECUC_BswM_00800]	Definition of EcucParamConfContainerDef BswMGeneral
[ECUC_BswM_00801]	Definition of EcucParamConfContainerDef BswMArbitration
[ECUC_BswM_00802]	Definition of EcucParamConfContainerDef BswMModeControl
[ECUC_BswM_00803]	Definition of EcucParamConfContainerDef BswMRteSwitch
[ECUC_BswM_00805]	Definition of EcucParamConfContainerDef BswMModeRequestPort
[ECUC_BswM_00806]	Definition of EcucParamConfContainerDef BswMRule
[ECUC_BswM_00807]	Definition of EcucParamConfContainerDef BswMModeCondition
[ECUC_BswM_00808]	Definition of EcucParamConfContainerDef BswMLogicalExpression
[ECUC_BswM_00809]	Definition of EcucParamConfContainerDef BswMActionList
[ECUC_BswM_00810]	Definition of EcucParamConfContainerDef BswMAction
[ECUC_BswM_00811]	Definition of EcucBooleanParamDef BswMDevErrorDetect
[ECUC_BswM_00813]	Definition of EcucFloatParamDef BswMMainFunctionPeriod
[ECUC_BswM_00814]	Definition of EcucEnumerationParamDef BswMLogicalOperator
[ECUC_BswM_00815]	Definition of EcucEnumerationParamDef BswMConditionType
[ECUC_BswM_00816]	Definition of EcucChoiceContainerDef BswMConditionValue

▽

△

Number	Heading
[ECUC_BswM_00817]	Definition of EcucReferenceDef BswMRuleTrueActionList
[ECUC_BswM_00818]	Definition of EcucReferenceDef BswMRuleFalseActionList
[ECUC_BswM_00819]	Definition of EcucReferenceDef BswMRuleExpressionRef
[ECUC_BswM_00820]	Definition of EcucChoiceReferenceDef BswMArgumentRef
[ECUC_BswM_00821]	Definition of EcucChoiceReferenceDef BswMConditionMode
[ECUC_BswM_00822]	Definition of EcucEnumerationParamDef BswMRequestProcessing
[ECUC_BswM_00823]	Definition of EcucParamConfContainerDef BswMActionListItem
[ECUC_BswM_00824]	Definition of EcucIntegerParamDef BswMActionListItemIndex
[ECUC_BswM_00825]	Definition of EcucChoiceReferenceDef BswMActionListItemRef
[ECUC_BswM_00826]	Definition of EcucChoiceContainerDef BswMAvailableActions
[ECUC_BswM_00827]	Definition of EcucParamConfContainerDef BswMLinScheduleSwitch
[ECUC_BswM_00828]	Definition of EcucParamConfContainerDef BswMPduGroupSwitch
[ECUC_BswM_00830]	Definition of EcucParamConfContainerDef BswMDeadlineMonitoringControl
[ECUC_BswM_00831]	Definition of EcucParamConfContainerDef BswMComMModeSwitch
[ECUC_BswM_00834]	Definition of EcucParamConfContainerDef BswMUserCallout
[ECUC_BswM_00837]	Definition of EcucParamConfContainerDef BswMNMControl
[ECUC_BswM_00838]	Definition of EcucEnumerationParamDef BswMNMAction
[ECUC_BswM_00840]	Definition of EcucEnumerationParamDef BswMComMRequestedMode
[ECUC_BswM_00841]	Definition of EcucReferenceDef BswMComMUserRef
[ECUC_BswM_00842]	Definition of EcucReferenceDef BswMLinScheduleRef
[ECUC_BswM_00843]	Definition of EcucStringParamDef BswMUserCalloutFunction
[ECUC_BswM_00849]	Definition of EcucReferenceDef BswMEnabledPduGroupRef

▽

△

Number	Heading
[ECUC_BswM_00850]	Definition of EcucReferenceDef BswMDisabledPduGroupRef
[ECUC_BswM_00851]	Definition of EcucReferenceDef BswMEnabledDMPduGroupRef
[ECUC_BswM_00852]	Definition of EcucReferenceDef BswMDisabledDMPduGroupRef
[ECUC_BswM_00853]	Definition of EcucParamConfContainerDef BswMPduRouterControl
[ECUC_BswM_00854]	Definition of EcucEnumerationParamDef BswMPduRouterAction
[ECUC_BswM_00855]	Definition of EcucReferenceDef BswMPduRoutingPathGroupRef
[ECUC_BswM_00856]	Definition of EcucChoiceContainerDef BswMModeRequestSource
[ECUC_BswM_00857]	Definition of EcucParamConfContainerDef BswMCanSMIndication
[ECUC_BswM_00858]	Definition of EcucParamConfContainerDef BswMFrSMIndication
[ECUC_BswM_00859]	Definition of EcucParamConfContainerDef BswMLinSMIndication
[ECUC_BswM_00860]	Definition of EcucParamConfContainerDef BswMEthSMIndication
[ECUC_BswM_00861]	Definition of EcucParamConfContainerDef BswMGenericRequest
[ECUC_BswM_00862]	Definition of EcucParamConfContainerDef BswMSwcModeRequest
[ECUC_BswM_00863]	Definition of EcucParamConfContainerDef BswMDcmComModeRequest
[ECUC_BswM_00864]	Definition of EcucForeignReferenceDef BswMModeValueRef
[ECUC_BswM_00866]	Definition of EcucStringParamDef BswMBswRequestedMode
[ECUC_BswM_00868]	Definition of EcucParamConfContainerDef BswMModeDeclaration
[ECUC_BswM_00869]	Definition of EcucParamConfContainerDef BswMBswMode
[ECUC_BswM_00870]	Definition of EcucReferenceDef BswMCanSMChannelRef
[ECUC_BswM_00871]	Definition of EcucReferenceDef BswMLinSMChannelRef
[ECUC_BswM_00872]	Definition of EcucReferenceDef BswMFrSMChannelRef
[ECUC_BswM_00873]	Definition of EcucReferenceDef BswMEthSMChannelRef

▽

△

Number	Heading
[ECUC_BswM_00874]	Definition of EcucIntegerParamDef BswMModeRequesterId
[ECUC_BswM_00880]	Definition of EcucParamConfContainerDef BswMComMIndication
[ECUC_BswM_00883]	Definition of EcucReferenceDef BswMComMChannelRef
[ECUC_BswM_00885]	Definition of EcucParamConfContainerDef BswMLinScheduleIndication
[ECUC_BswM_00888]	Definition of EcucEnumerationParamDef BswMRuleInitState
[ECUC_BswM_00890]	Definition of EcucParamConfContainerDef BswMNvMRequest
[ECUC_BswM_00891]	Definition of EcucReferenceDef BswMNvMBlockRef
[ECUC_BswM_00892]	Definition of EcucParamConfContainerDef BswMSwcModeNotification
[ECUC_BswM_00893]	Definition of EcucForeignReferenceDef BswMSwcModeNotificationMode DeclarationGroupPrototypeRef
[ECUC_BswM_00894]	Definition of EcucEnumerationParamDef BswMActionListExecution
[ECUC_BswM_00895]	Definition of EcucParamConfContainerDef BswMConfig
[ECUC_BswM_00896]	Definition of EcucForeignReferenceDef BswMSwitchedMode
[ECUC_BswM_00899]	Definition of EcucParamConfContainerDef BswMSchMSwitch
[ECUC_BswM_00901]	Definition of EcucForeignReferenceDef BswMSchMSwitchedMode
[ECUC_BswM_00902]	Definition of EcucBooleanParamDef BswMAbortOnFail
[ECUC_BswM_00904]	Definition of EcucParamConfContainerDef BswMEcuMWakeupSource
[ECUC_BswM_00905]	Definition of EcucReferenceDef BswMEcuMWakeupSrcRef
[ECUC_BswM_00906]	Definition of EcucParamConfContainerDef BswMTriggerIPduSend
[ECUC_BswM_00907]	Definition of EcucReferenceDef BswMTriggeredIPduRef
[ECUC_BswM_00908]	Definition of EcucParamConfContainerDef BswMComMModeLimitation
[ECUC_BswM_00909]	Definition of EcucParamConfContainerDef BswMComMAllowCom
[ECUC_BswM_00910]	Definition of EcucBooleanParamDef BswMComMLimitMode

▽

△

Number	Heading
[ECUC_BswM_00911]	Definition of EcucReferenceDef BswMComMLimitChannelRef
[ECUC_BswM_00912]	Definition of EcucReferenceDef BswMComMAllowChannelRef
[ECUC_BswM_00913]	Definition of EcucBooleanParamDef BswMPduGroupSwitchReinit
[ECUC_BswM_00914]	Definition of EcucParamConfContainerDef BswMLinTpModeRequest
[ECUC_BswM_00915]	Definition of EcucReferenceDef BswMLinTpChannelRef
[ECUC_BswM_00918]	Definition of EcucBooleanParamDef BswMComAllowed
[ECUC_BswM_00922]	Definition of EcucParamConfContainerDef BswMComMPncRequest
[ECUC_BswM_00924]	Definition of EcucReferenceDef BswMComMPncRef
[ECUC_BswM_00925]	Definition of EcucParamConfContainerDef BswMDcmApplicationUpdated Indication
[ECUC_BswM_00926]	Definition of EcucParamConfContainerDef BswMBswModeNotification
[ECUC_BswM_00927]	Definition of EcucForeignReferenceDef BswMBswModeDeclarationGroup PrototypeRef
[ECUC_BswM_00928]	Definition of EcucParamConfContainerDef BswMModeInitValue
[ECUC_BswM_00932]	Definition of EcucStringParamDef BswMBswModeInitValue
[ECUC_BswM_00935]	Definition of EcucBooleanParamDef BswMNestedExecutionOnly
[ECUC_BswM_00936]	Definition of EcucParamConfContainerDef BswMDataTypeMappingSets
[ECUC_BswM_00937]	Definition of EcucForeignReferenceDef BswMDataTypeMappingSetRef
[ECUC_BswM_00938]	Definition of EcucBooleanParamDef BswMCanSMEnabled
[ECUC_BswM_00939]	Definition of EcucBooleanParamDef BswMComMEnabled
[ECUC_BswM_00940]	Definition of EcucBooleanParamDef BswMDcmEnabled
[ECUC_BswM_00941]	Definition of EcucBooleanParamDef BswMEcuMEnabled
[ECUC_BswM_00942]	Definition of EcucBooleanParamDef BswMEthSMEnabled
[ECUC_BswM_00943]	Definition of EcucBooleanParamDef BswMFrSMEnabled

▽

△

Number	Heading
[ECUC_BswM_00944]	Definition of EcucBooleanParamDef BswMLinSMEnabled
[ECUC_BswM_00945]	Definition of EcucBooleanParamDef BswMLinTPEnabled
[ECUC_BswM_00946]	Definition of EcucBooleanParamDef BswMNvMEnabled
[ECUC_BswM_00949]	Definition of EcucBooleanParamDef BswMGenericRequestEnabled
[ECUC_BswM_00950]	Definition of EcucParamConfContainerDef BswMSwitchPort
[ECUC_BswM_00951]	Definition of EcucForeignReferenceDef BswMModeSwitchInterfaceRef
[ECUC_BswM_00952]	Definition of EcucReferenceDef BswMRteSwitchPortRef
[ECUC_BswM_00954]	Definition of EcucParamConfContainerDef BswMUserIncludeFiles
[ECUC_BswM_00955]	Definition of EcucStringParamDef BswMUserIncludeFile
[ECUC_BswM_00956]	Definition of EcucParamConfContainerDef BswMNvMJobModeIndication
[ECUC_BswM_00957]	Definition of EcucEnumerationParamDef BswMNvmService
[ECUC_BswM_00958]	Definition of EcucParamConfContainerDef BswMSwitchIPduMode
[ECUC_BswM_00959]	Definition of EcucReferenceDef BswMSwitchIPduModeRef
[ECUC_BswM_00960]	Definition of EcucBooleanParamDef BswMSwitchIPduModeValue
[ECUC_BswM_00961]	Definition of EcucParamConfContainerDef BswMEcuMSelectShutdown Target
[ECUC_BswM_00962]	Definition of EcucReferenceDef BswMEcuMSleepModeRef
[ECUC_BswM_00963]	Definition of EcucParamConfContainerDef BswMEcuMGoDownHaltPoll
[ECUC_BswM_00964]	Definition of EcucReferenceDef BswMEcuMUserIdRef
[ECUC_BswM_00965]	Definition of EcucBooleanParamDef BswMJ1939NmEnabled
[ECUC_BswM_00966]	Definition of EcucParamConfContainerDef BswMJ1939NmIndication
[ECUC_BswM_00967]	Definition of EcucReferenceDef BswMJ1939NmChannelRef
[ECUC_BswM_00969]	Definition of EcucReferenceDef BswMDcmComMChannelRef

▽

△

Number	Heading
[ECUC_BswM_00974]	Definition of EcucParamConfContainerDef BswMSdClientServiceMode Request
[ECUC_BswM_00984]	Definition of EcucReferenceDef BswMPartitionRef
[ECUC_BswM_00985]	Definition of EcucParamConfContainerDef BswMJ1939DcmBroadcastStatus
[ECUC_BswM_00987]	Definition of EcucBooleanParamDef BswMJ1939DcmEnabled
[ECUC_BswM_00988]	Definition of EcucReferenceDef BswMJ1939DcmChannelRef
[ECUC_BswM_00993]	Definition of EcucEnumerationParamDef BswMEcuMShutdownTarget
[ECUC_BswM_00994]	Definition of EcucReferenceDef BswMEcuMResetModeRef
[ECUC_BswM_00997]	Definition of EcucReferenceDef BswMJ1939NmNodeRef
[ECUC_BswM_00998]	Definition of EcucParamConfContainerDef BswMJ1939RmStateSwitch
[ECUC_BswM_00999]	Definition of EcucReferenceDef BswMComMNetworkHandleRef
[ECUC_BswM_01000]	Definition of EcucReferenceDef BswMJ1939RmChannelRef
[ECUC_BswM_01001]	Definition of EcucReferenceDef BswMJ1939RmNodeRef
[ECUC_BswM_01002]	Definition of EcucEnumerationParamDef BswMJ1939RmRequestedState
[ECUC_BswM_01004]	Definition of EcucParamConfContainerDef BswMSdConsumedEventGroup ModeRequest
[ECUC_BswM_01005]	Definition of EcucParamConfContainerDef BswMSdServerServiceMode Request
[ECUC_BswM_01007]	Definition of EcucReferenceDef BswMSdServerMethodsRef
[ECUC_BswM_01008]	Definition of EcucReferenceDef BswMSdEventHandlerRef
[ECUC_BswM_01009]	Definition of EcucReferenceDef BswMSdClientMethodsRef
[ECUC_BswM_01010]	Definition of EcucReferenceDef BswMSdConsumedEventGroupRef
[ECUC_BswM_01011]	Definition of EcucParamConfContainerDef BswMSdClientServiceCurrent State
[ECUC_BswM_01012]	Definition of EcucParamConfContainerDef BswMSdConsumedEventGroup CurrentState
[ECUC_BswM_01013]	Definition of EcucParamConfContainerDef BswMSdEventHandlerCurrent State

▽

△

Number	Heading
[ECUC_BswM_01014]	Definition of EcucParamConfContainerDef BswMComMInitiateReset
[ECUC_BswM_01015]	Definition of EcucEnumerationParamDef BswMSdServerServiceState
[ECUC_BswM_01016]	Definition of EcucEnumerationParamDef BswMSdClientServiceState
[ECUC_BswM_01017]	Definition of EcucEnumerationParamDef BswMSdConsumedEventGroup State
[ECUC_BswM_01021]	Definition of EcucParamConfContainerDef BswMRteModeRequest
[ECUC_BswM_01022]	Definition of EcucParamConfContainerDef BswMRteModeRequestPort
[ECUC_BswM_01023]	Definition of EcucReferenceDef BswMRteModeRequestPortRef
[ECUC_BswM_01024]	Definition of EcucForeignReferenceDef BswMRequestedModeRef
[ECUC_BswM_01025]	Definition of EcucInstanceReferenceDef BswMRteModeRequestPort InterfaceRef
[ECUC_BswM_01028]	Definition of EcucReferenceDef BswMLinSMChannelRef
[ECUC_BswM_01031]	Definition of EcucForeignReferenceDef BswMSchMModeDeclarationGroup Ref
[ECUC_BswM_01032]	Definition of EcucParamConfContainerDef BswMJ1939DcmStateSwitch
[ECUC_BswM_01033]	Definition of EcucReferenceDef BswMJ1939DcmChannelRef
[ECUC_BswM_01034]	Definition of EcucReferenceDef BswMJ1939DcmNodeRef
[ECUC_BswM_01035]	Definition of EcucEnumerationParamDef BswMJ1939DcmRequestedState
[ECUC_BswM_01036]	Definition of EcucBooleanParamDef BswMPduRouterDisableInitBuffer
[ECUC_BswM_01037]	Definition of EcucParamConfContainerDef BswMFrSMAllSlots
[ECUC_BswM_01038]	Definition of EcucReferenceDef BswMFrSMAllSlotsNetworkHandleRef
[ECUC_BswM_01039]	Definition of EcucParamConfContainerDef BswMCompuScaleModeValue
[ECUC_BswM_01040]	Definition of EcucForeignReferenceDef BswMCompuMethodRef
[ECUC_BswM_01041]	Definition of EcucStringParamDef BswMCompuConstText
[ECUC_BswM_01044]	Definition of EcucEnumerationParamDef BswMEcuMState

▽

△

Number	Heading
[ECUC_BswM_01045]	Definition of EcucParamConfContainerDef BswMEcuMStateSwitch
[ECUC_BswM_01046]	Definition of EcucForeignReferenceDef BswMSwcModeRequestVariable DataPrototypeRef
[ECUC_BswM_01047]	Definition of EcucBooleanParamDef BswMSdEnabled
[ECUC_BswM_01049]	Definition of EcucReferenceDef BswMNmChannelRef
[ECUC_BswM_01050]	Definition of EcucIntegerParamDef BswMReportFailRuntimeErrorId
[ECUC_BswM_01052]	Definition of EcucParamConfContainerDef BswMEventRequestPort
[ECUC_BswM_01053]	Definition of EcucChoiceContainerDef BswMEventRequestSource
[ECUC_BswM_01054]	Definition of EcucParamConfContainerDef BswMClearEventRequest
[ECUC_BswM_01055]	Definition of EcucReferenceDef BswMClearEventRequestPortRef
[ECUC_BswM_01056]	Definition of EcucEnumerationParamDef BswMEventRequestProcessing
[ECUC_BswM_01057]	Definition of EcucForeignReferenceDef BswMRteModeRequestVariableData PrototypeSRRef
[ECUC_BswM_01058]	Definition of EcucParamConfContainerDef BswMTimer
[ECUC_BswM_01059]	Definition of EcucParamConfContainerDef BswMTimerControl
[ECUC_BswM_01060]	Definition of EcucEnumerationParamDef BswMTimerAction
[ECUC_BswM_01061]	Definition of EcucFloatParamDef BswMTimerValue
[ECUC_BswM_01062]	Definition of EcucReferenceDef BswMTimerRef
[ECUC_BswM_01063]	Definition of EcucModuleDef BswM
[ECUC_BswM_01064]	Definition of EcucParamConfContainerDef BswMEcuMDriverInitListBswM
[ECUC_BswM_01065]	Definition of EcucReferenceDef BswMEcuMDriverInitListBswMRef
[ECUC_BswM_01066]	Definition of EcucParamConfContainerDef BswMEthIfPortGroupLinkState Chg
[ECUC_BswM_01067]	Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef

▽

△

Number	Heading
[ECUC_BswM_01068]	Definition of EcucParamConfContainerDef BswMEthIfSwitchPortGroup RequestMode
[ECUC_BswM_01069]	Definition of EcucReferenceDef BswMEthIfSwitchPortGroupRef
[ECUC_BswM_01070]	Definition of EcucEnumerationParamDef BswMEthTrcvMode
[ECUC_BswM_01071]	Definition of EcucBooleanParamDef BswMNmEnabled
[ECUC_BswM_01072]	Definition of EcucBooleanParamDef BswMEthIfEnabled
[ECUC_BswM_01075]	Definition of EcucParamConfContainerDef BswMNmCarWakeUpIndication
[ECUC_BswM_01076]	Definition of EcucIntegerParamDef BswMActionListPriority
[ECUC_BswM_01077]	Definition of EcucParamConfContainerDef BswMSdServiceGroupSwitch
[ECUC_BswM_01078]	Definition of EcucReferenceDef BswMDisabledSdServiceGroupRef
[ECUC_BswM_01079]	Definition of EcucReferenceDef BswMEnabledSdServiceGroupRef
[ECUC_BswM_01080]	Definition of EcucReferenceDef BswMSchMSwitchPortRef
[ECUC_BswM_01081]	Definition of EcucParamConfContainerDef BswMNmStateChange Notification
[ECUC_BswM_01082]	Definition of EcucReferenceDef BswMNmChannelRef
[ECUC_BswM_01083]	Definition of EcucParamConfContainerDef BswMSwitchAckNotification
[ECUC_BswM_01084]	Definition of EcucReferenceDef BswMSwitchPortRef
[ECUC_BswM_01085]	Definition of EcucParamConfContainerDef BswMEcuMIndication
[ECUC_BswM_01086]	Definition of EcucParamConfContainerDef BswMEcuMRUNRequest Indication
[ECUC_BswM_01087]	Definition of EcucEnumerationParamDef BswMEcuMRUNRequestProtocol Port
[ECUC_BswM_01088]	Definition of EcucParamConfContainerDef BswMEthIfStartAllPorts
[ECUC_BswM_01089]	Definition of EcucParamConfContainerDef BswMIdsMBlockStateChange Request
[ECUC_BswM_01090]	Definition of EcucReferenceDef BswMIdsMBlockStateChangeRequestRef
[ECUC_BswM_01091]	Definition of EcucParamConfContainerDef BswMSoAdSoConModeChg

▽

△

Number	Heading
[ECUC_BswM_01092]	Definition of EcucReferenceDef BswMSoAdSocketConnectionRef
[ECUC_BswM_01093]	Definition of EcucBooleanParamDef BswMSdControlEnabled
[ECUC_BswM_01094]	Definition of EcucParamConfContainerDef BswMSetFirewallState
[ECUC_BswM_01095]	Definition of EcucParamConfContainerDef BswMIdsMTransmissionState ChangeRequest
[ECUC_BswM_01096]	Definition of EcucEnumerationParamDef BswMIdsMTransmissionRequested State
[ECUC_BswM_01097]	Definition of EcucBooleanParamDef BswMSoAdEnabled
[SWS_BSWM_91010]	Definition of SenderReceiverInterface ClientServiceRequest_{Arb Name}_{PortName}
[SWS_BswM_00001]	Definition of imported datatypes of module BswM
[SWS_BswM_00008]	Definition of optional interfaces requested by module BswM
[SWS_BswM_00055]	
[SWS_BswM_00283]	API to Request Port Mappings: This table describes which request port shall be triggered when an API is called. In addition, it describes which configuration parameter(s) correspond to which API parameter(s)

Table B.12: Changed Specification Items in R26-11

B.1.21 Deleted Specification Items in R26-11

Number	Heading
[ECUC_BswM_00812]	Definition of EcucBooleanParamDef BswMVersionInfoApi
[ECUC_BswM_00970]	Definition of EcucParamConfContainerDef BswMCoreHaltMode
[ECUC_BswM_00971]	Definition of EcucReferenceDef BswMTargetCoreRef
[ECUC_BswM_00972]	Definition of EcucStringParamDef BswMCoreHaltActivationState
[ECUC_BswM_00989]	Definition of EcucParamConfContainerDef BswMPartitionRestarted
[ECUC_BswM_00990]	Definition of EcucParamConfContainerDef BswMModeSwitchErrorEvent
[ECUC_BswM_01030]	Definition of EcucReferenceDef BswMRteSwitchPortRef
[ECUC_BswM_01073]	Definition of EcucParamConfContainerDef BswMRteStart

▽

△

Number	Heading
[ECUC_BswM_01074]	Definition of EcucParamConfContainerDef BswMRteStop
[SWS_BswM_00003]	Definition of API function BswM_GetVersionInfo
[SWS_BswM_00193]	Definition of API function BswM_BswMPartitionRestarted
[SWS_BswM_00259]	
[SWS_BswM_00271]	
[SWS_BswM_00272]	

Table B.13: Deleted Specification Items in R26-11

B.1.22 Added Constraints in R26-11

none

B.1.23 Changed Constraints in R26-11

Number	Heading
[SWS_BswM_CONSTR_00006]	

Table B.14: Changed Constraints in R26-11

B.1.24 Deleted Constraints in R26-11

none