Document Title	Specification of Basic Software Mode Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	313
Document Classification	Standard
Document Version	1.5.0
Document Status	Final
Part of Release	4.1
Revision	3

Document Change History			
Date	**Version**	**Changed by**	**Change Description**
31.03.2014	1.5.0	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
31.10.2013	1.4.0	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
28.02.2013	1.3.0	AUTOSAR Administration	• Extended to support Pretended Networking mode handling • Adaption of BswM to concept Enhanced BSW Allocation • Extended to support Heavy Duty Vehicles and J1939
09.12.2011	1.2.0	AUTOSAR Administration	• Support of Mode Machine Instances assigned to the SchM • Include of user defined header files • Possibility to provide an initial value for a BswMModeRequestPort
22.10.2010	1.1.0	AUTOSAR Administration	• Include file BswMUserCallout.h added. This user defined header file contains declarations of the call out functions. • Requirement that the BswM module

Document Change History			
Date	**Version**	**Changed by**	**Change Description**
			shall perform inter module version checks added • Information added for each configurable action which API to call • Functions BswM_TriggerSlaveRTEStop and BswM_TriggerStartUpPhase2 added to control the start and stop of the RTE on slave cores
30.11.2009	1.0.0	AUTOSAR Administration	Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

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

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

Table of Contents

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module BSW Mode Manager (BswM).
The BSW Mode Manager is the module that implements the part of the Vehicle Mode Management and Application Mode Management concept that resides in the BSW. Its responsibility is to arbitrate mode requests from application layer SW-Cs or other BSW modules based on simple rules, and perform actions based on the arbitration result.

2 Acronyms and abbreviations

Abbreviation / Acronym:	*Description:*
AMM	Application Mode Management
BSW	Basic Software
BswM	BSW Mode Manager
Dem	Diagnostic Event Manager
Det	Development Error Tracer
ECU	Electronic Control Unit
ICOM	Intelligent Communication Controller
RTE	Real Time Environment
VMM	Vehicle Mode Management

Table 1: Table of acronyms and abbreviations

3 Related documentation

3.1 Input documents

[1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf.pdf

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

[4] Requirements on Mode Management
AUTOSAR_SRS_ModeManagement.pdf

[5] Specification of Communication
AUTOSAR_SWS_COM.pdf

[6] Specification of FlexRay State Manager
AUTOSAR_SWS_FlexRayStateManager.pdf

[7] Specification of PDU Router
AUTOSAR_SWS_PDURouter.pdf

[8] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[9] Specification of Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[10] Specification of RTE Software
AUTOSAR_SWS_RTE.pdf

[11] Specification of Diagnostic Communication Manager
AUTOSAR_SWS_DiagnosticCommunicationManager.pdf

[12] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

[13] Specification of LIN State Manager
AUTOSAR_SWS_LINStateManager.pdf

[14] Specification of CAN State Manager
AUTOSAR_SWS_CANStateManager.pdf

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

[16] Specification of Communication Manager
AUTOSAR_SWS_COMManager.pdf

[17] Specification of Ethernet State Manager
AUTOSAR_SWS_EthernetStateManager.pdf

[18] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

3.2 Related standards and norms

None.

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [18] (SWS BSW General), which is also valid for BSW Mode Manager.

Thus, the specification SWS BSW General shall be considered as an 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: AUTOSAR_EXP_ModemanagementGuide.pdf

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.

5.1 RTE

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

5.2 EcuM - Fixed

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

5.3 EcuM - Flex

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

5.4 WdgM

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

5.5 ComM

Mode Switch Indications originating from the ComM go through the BswM for further propagation to the SW-Cs.
The BswM can request communication modes at the ComM by means of ComMUsers.

5.6 COM

The handling of I-PDU Groups in COM is performed by the BswM. As a part of I-PDU group start/stop, it is possible to have the included signal values reset to their corresponding initialization values.
BswM handles the enabling and disabling of deadline monitoring of signals in COM.
BswM can also trigger transmission of an I-PDU.

5.7 PduR

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

router.

5.8 CanSM

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

5.9 LinSM

BswM coordinates switching of LIN Schedule Tables in the LinSM with starting and stopping of the corresponding I-PDU groups in COM.
Mode Switch Indications originating from the LinSM go through the BswM for further propagation to the SW-Cs.

5.10 LinTp

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

5.11 FrSM

Mode Switch Indications originating from the FrSM go through the BswM for further propagation to the SW-Cs.
The usage of “Single Slot Mode” on FlexRay is controlled by the FrSM by request of BswM. The send capability of the FlexRay stack can be controlled by the BswM via FrSM by calling the API FrSM_SetEcuPassive.

5.12 EthSM

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

5.13 DCM

The DCM performs mode requests to the BswM based on the diagnostic requests it receives.
Example: DCM can request “Disable Normal Communication”. During this mode BswM will turn off the corresponding I-PDU groups and NM PDUs.

5.14 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.

5.15 J1939Nm

The J1939Nm provides a state indication via BswM_J1939Nm_StateChangeNotification.

5.16 J1939Rm

BswM changes states of J1939Rm via J1939Rm_SetState.

5.17 NM Interface

BswM will use the Nm_EnableCommunication and Nm_DisableCommunication to control the NM communication based on the current mode.
Example: In “Disable Normal Communication” mode, the BswM needs to disable NM communication on the corresponding NM channel.

5.18 NvM

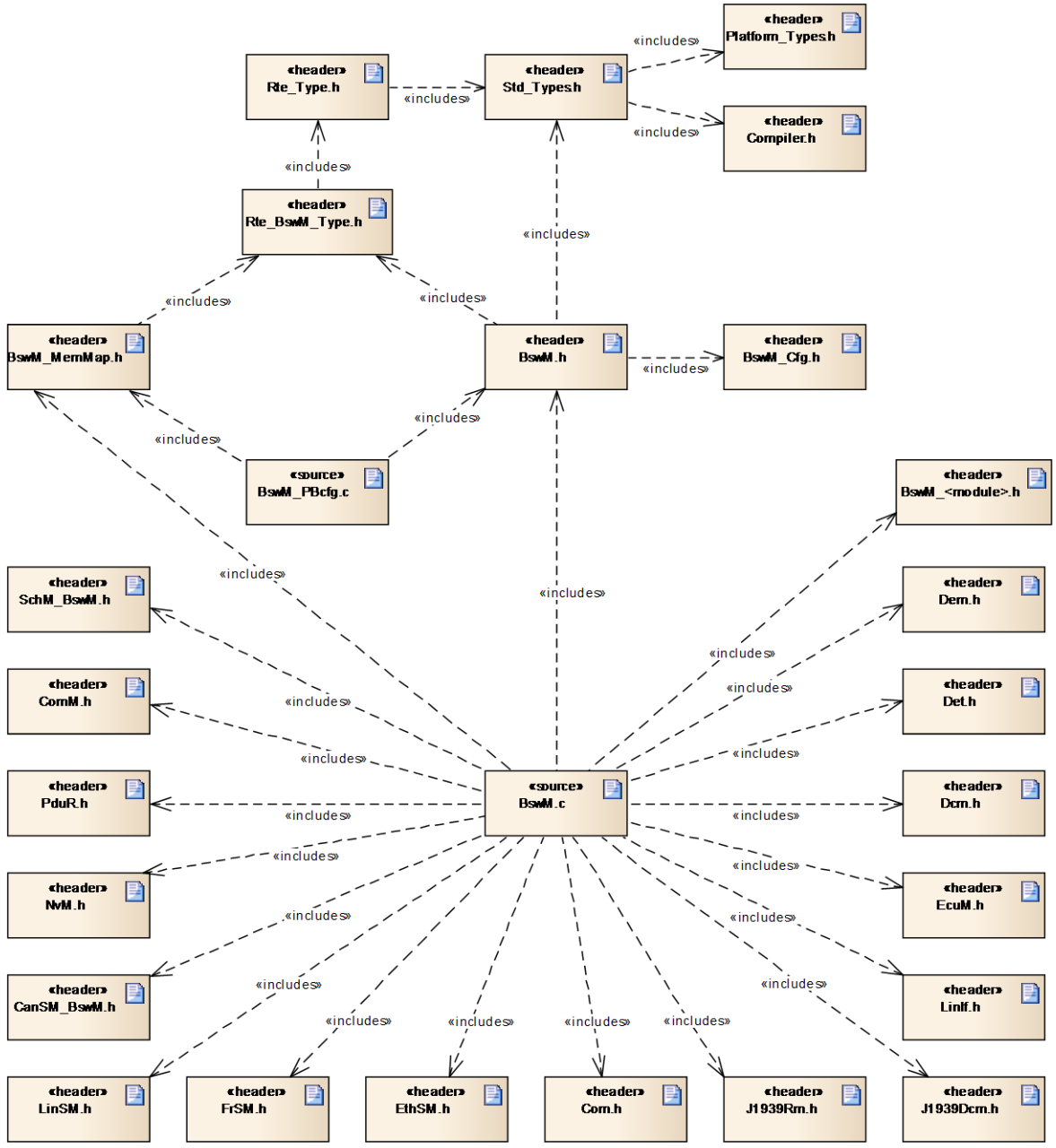
The NvM module reports the state of its blocks to the BswM via “integration code“ registered as NvM callbacks. BswM has actions causing the NvM to read and write all blocks during startup and shutdown.

5.19 OS

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

5.20 File structure

[SWS_BswM_00218]⌜



⌟()
Figure 1: File structure of BSW Mode Manager

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

5.20.1 Code file structure

[SWS_BswM_00024]「
The code file structure shall not be defined within this specification completely. At this point it shall be pointed out that the code-file structure shall include the following files named:

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

These files shall contain all link time and post-build time configurable parameters.」
(SRS_BSW_00380, SRS_BSW_00419)

5.20.2 Header file structure

[SWS_BswM_00025]「
The BswM shall include the header files of all other BSW modules which API functions it uses.
Specifically it shall include StdTypes.h and ComStack_Types.h to avoid redefinition of types.
」()

[SWS_BswM_00026]「
The BswM module shall provide the following set of header files for inclusion in other BSW modules only if the relevant configuration parameter is set to true. The header file shall provide interfaces and the corresponding types relevant to the other BSW module:

1. BswM header file: BswM.h, BswM_CanSM.h, BswM_LinSM.h, BswMLinTp.h, BswM_FrSM.h, BswM_EthSM.h, BswM_EcuM.h, BswM_ComM.h, BswM_WdgM.h, BswM_DCM.h, BswM_Sd.h, BswM_J1939Nm.h, BswM_J1939Dcm.h
2. BswM configuration file: BswM_Cfg.h」

(SRS_BSW_00381, SRS_BSW_00412, SRS_BSW_00415)

6 Requirements traceability

Requirement	Description	Satisfied by
-	-	SWS_BswM_00001
-	-	SWS_BswM_00007
-	-	SWS_BswM_00008
-	-	SWS_BswM_00017
-	-	SWS_BswM_00018
-	-	SWS_BswM_000195
-	-	SWS_BswM_00019
-	-	SWS_BswM_00025
-	-	SWS_BswM_00035
-	-	SWS_BswM_00037
-	-	SWS_BswM_00041
-	-	SWS_BswM_00042
-	-	SWS_BswM_00043
-	-	SWS_BswM_00044
-	-	SWS_BswM_00045
-	-	SWS_BswM_00053
-	-	SWS_BswM_00054
-	-	SWS_BswM_00055
-	-	SWS_BswM_00056
-	-	SWS_BswM_00058
-	-	SWS_BswM_00059
-	-	SWS_BswM_00060
-	-	SWS_BswM_00061
-	-	SWS_BswM_00062
-	-	SWS_BswM_00064
-	-	SWS_BswM_00066
-	-	SWS_BswM_00067
-	-	SWS_BswM_00068
-	-	SWS_BswM_00069
-	-	SWS_BswM_00075
-	-	SWS_BswM_00076
-	-	SWS_BswM_00088
-	-	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_00104
-	-	SWS_BswM_00110
-	-	SWS_BswM_00113
-	-	SWS_BswM_00115
-	-	SWS_BswM_00116
-	-	SWS_BswM_00117
-	-	SWS_BswM_00118
-	-	SWS_BswM_00119
-	-	SWS_BswM_00120
-	-	SWS_BswM_00121
-	-	SWS_BswM_00128
-	-	SWS_BswM_00129
-	-	SWS_BswM_00131
-	-	SWS_BswM_00133
-	-	SWS_BswM_00137
-	-	SWS_BswM_00138
-	-	SWS_BswM_00139
-	-	SWS_BswM_00140
-	-	SWS_BswM_00141
-	-	SWS_BswM_00142
-	-	SWS_BswM_00143
-	-	SWS_BswM_00144
-	-	SWS_BswM_00145
-	-	SWS_BswM_00147
-	-	SWS_BswM_00150
-	-	SWS_BswM_00152
-	-	SWS_BswM_00154
-	-	SWS_BswM_00156
-	-	SWS_BswM_00157
-	-	SWS_BswM_00160
-	-	SWS_BswM_00164
-	-	SWS_BswM_00165
-	-	SWS_BswM_00166

-	-	SWS_BswM_00187
-	-	SWS_BswM_00188
-	-	SWS_BswM_00189
-	-	SWS_BswM_00190
-	-	SWS_BswM_00191
-	-	SWS_BswM_00192
-	-	SWS_BswM_00193
-	-	SWS_BswM_00194
-	-	SWS_BswM_00196
-	-	SWS_BswM_00197
-	-	SWS_BswM_00198
-	-	SWS_BswM_00199
-	-	SWS_BswM_00200
-	-	SWS_BswM_00201
-	-	SWS_BswM_00202
-	-	SWS_BswM_00203
-	-	SWS_BswM_00204
-	-	SWS_BswM_00206
-	-	SWS_BswM_00207
-	-	SWS_BswM_00209
-	-	SWS_BswM_00210
-	-	SWS_BswM_00212
-	-	SWS_BswM_00213
-	-	SWS_BswM_00214
-	-	SWS_BswM_00215
-	-	SWS_BswM_00216
-	-	SWS_BswM_00217
-	-	SWS_BswM_00218
-	-	SWS_BswM_00219
-	-	SWS_BswM_00220
-	-	SWS_BswM_00221
-	-	SWS_BswM_00222
BSW009176	-	SWS_BswM_00038
SRS_BSW_00101	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	SWS_BswM_00002
SRS_BSW_00170	The AUTOSAR SW Components shall provide information about their dependency from faults,	SWS_BswM_09999

	signal qualities, driver demands	
SRS_BSW_00336	Basic SW module shall be able to shutdown	SWS_BswM_09999
SRS_BSW_00339	Reporting of production relevant error status	SWS_BswM_09999
SRS_BSW_00380	Configuration parameters being stored in memory shall be placed into separate c-files	SWS_BswM_00024
SRS_BSW_00381	The pre-compile time parameters shall be placed into a separate configuration header file	SWS_BswM_00026
SRS_BSW_00387	The Basic Software Module specifications shall specify how the callback function is to be implemented	SWS_BswM_09999
SRS_BSW_00399	Parameter-sets shall be located in a separate segment and shall be loaded after the code	SWS_BswM_09999
SRS_BSW_00400	Parameter shall be selected from multiple sets of parameters after code has been loaded and started	SWS_BswM_09999
SRS_BSW_00405	BSW Modules shall support multiple configuration sets	SWS_BswM_09999
SRS_BSW_00406	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	SWS_BswM_00077, SWS_BswM_00078, SWS_BswM_00079, SWS_BswM_00080, SWS_BswM_00081, SWS_BswM_00082, SWS_BswM_00083, SWS_BswM_00084, SWS_BswM_00086, SWS_BswM_00109, SWS_BswM_00112, SWS_BswM_00132, SWS_BswM_00134, SWS_BswM_00149, SWS_BswM_00153, SWS_BswM_00159, SWS_BswM_00161, SWS_BswM_00205, SWS_BswM_00208, SWS_BswM_00211
SRS_BSW_00407	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	SWS_BswM_00003
SRS_BSW_00409	All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration	SWS_BswM_09999
SRS_BSW_00412	References to c-	SWS_BswM_00026

	configuration parameters shall be placed into a separate h-file	
SRS_BSW_00415	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	SWS_BswM_00026
SRS_BSW_00419	If a pre-compile time configuration parameter is implemented as "const" it should be placed into a separate c-file	SWS_BswM_00024
SRS_ModeMgm_09174	The BSW Mode Manager shall support the 'disable normal Communication'	SWS_BswM_00038
SRS_ModeMgm_09175	A configurable Set of Mode dependent enabled and concomitant disabled IPDU groups shall be supported	SWS_BswM_00038
SRS_ModeMgm_09177	The rules of the mode arbitration shall be pre-compile configurable	SWS_BswM_00010, SWS_BswM_00012, SWS_BswM_00015, SWS_BswM_00016
SRS_ModeMgm_09179	The BSW Mode Manager shall provide an Interface to allow Mode Requests of SWC's	SWS_BswM_00046
SRS_ModeMgm_09180	The BSW Mode Manager shall evaluate the current mode requests	SWS_BswM_00009, SWS_BswM_00013, SWS_BswM_00014
SRS_ModeMgm_09182	The BSW Mode Manager shall propagate a performed mode change to all local SW-Cs	SWS_BswM_00038
SRS_ModeMgm_09183	Configurable Mode Activation initiated Reset of Signals to Initial Values shall be supported	SWS_BswM_00038
SRS_ModeMgm_09184	The mode manager shall be able to use a COM interface to activate, respectively deactivate, I-PDU groups	SWS_BswM_00038
SRS_ModeMgm_09228	The BSW Mode Manager shall provide an Interface to allow Mode Requests of BSW Modules	SWS_BswM_00046, SWS_BswM_00047, SWS_BswM_00048, SWS_BswM_00049, SWS_BswM_00050, SWS_BswM_00051, SWS_BswM_00052, SWS_BswM_00148, SWS_BswM_00158
SRS_ModeMgm_09229	The mode manager shall be able to make generic, configured callouts of void functions to other BSW	SWS_BswM_00039, SWS_BswM_00040

	modules	
SRS_ModeMgm_09230	All actions shall only be performed on mode change	SWS_BswM_00011, SWS_BswM_00023

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

Requirement	***Satisfied by***
[SRS_BSW_00344] Reference to link--time configuration	SWS_BswM_00021
[SRS_BSW_00404] Reference to post build time configuration	Chapter 5
[SRS_BSW_00405] Reference to multiple configuration sets	Not applicable
[SRS_BSW_00345] Configuration at Compile time	SWS_BswM_00020
[SRS_BSW_00159] Automatic configuration	Chapter 10
[SRS_BSW_00167] Static configuration checking	Chapter 10
[SRS_BSW_00171] Configurability of optional functionality	Chapter 10
[SRS_BSW_00170] Data for reconfiguration of AUTOSAR SW--Components	Not applicable
[SRS_BSW_00380] Separate C--Files for configuration parameters	[SWS_BswM_00024
[SRS_BSW_00419] Separate C--Files for pre--compile time configuration parameters	[SWS_BswM_00024
[SRS_BSW_00381] Separate configuration header file for pre--compile time parameters	[SWS_BswM_00026
[SRS_BSW_00412] Separate H--File for configuration parameters	[SWS_BswM_00026
[SRS_BSW_00383] List dependencies of configuration files	Chapter 5
[SRS_BSW_00384] List dependencies to other modules	Chapter 5
[SRS_BSW_00387] Specify the configuration class of callback function	Not applicable
[SRS_BSW_00388] Introduce containers	Chapter 10
[SRS_BSW_00389] Containers shall have names	Chapter 10
[SRS_BSW_00390] Parameter content shall be unique within the module	Chapter 10
[SRS_BSW_00391] Parameter shall have unique names	Chapter 10

[SRS_BSW_00392] Parameters shall have a type	Chapter 10
[SRS_BSW_00393] Parameters shall have a range	Chapter 10
[SRS_BSW_00394] Specify the scope of the parameters	Chapter 10
[SRS_BSW_00395] List the required parameters (per parameter)	Chapter 10
[SRS_BSW_00396] Configuration classes	Chapter 10
[SRS_BSW_00397] Pre--compile--time parameters	Chapter 10
[SRS_BSW_00398] Link--time parameters	Chapter 10
[SRS_BSW_00399] Loadable Post--build time parameters	Not applicable
[SRS_BSW_00400] Selectable Post--build time parameters	Not applicable
[SRS_BSW_00438] Post Build Configuration Data Structure	Chapter 10
[SRS_BSW_00402] Published information	Chapter 10
[SRS_BSW_00101] Initialization interface	[SWS_BswM_00002
[SRS_BSW_00406] Check module initialization	[SWS_BswM_00077, The corresponding configuration container for this API is BswMComMIndication. [SWS_BswM_00078, 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. [SWS_BswM_00079, The corresponding configuration container for this API is BswMCanSMIndication. [SWS_BswM_00080, The corresponding configuration container for this API is BswMEthSMIndication. [SWS_BswM_00081, The corresponding configuration container for this API is BswMFrSMIndication. [SWS_BswM_00082, The corresponding configuration container for this API is BswMLinSMIndication.

	[SWS_BswM_00083, The corresponding configuration container for this API is BswMEcuMIndication. [SWS_BswM_00084, The corresponding configuration container for this API is BswMLinScheduleIndication. [SWS_BswM_00086
[SRS_BSW_00407] Function to read out published parameters	[SWS_BswM_00003
[SRS_BSW_00423] Usage of SW--C template to describe BSW modules with AUTOSAR Interfaces	Chapter 7.7
[SRS_BSW_00336] Shutdown interface	Not Applicable
[SRS_BSW_00337] Classification of errors	Chapter 7.3
[SRS_BSW_00338] Detection and Reporting of development errors	Chapter 7.4
[SRS_BSW_00369] Do not return development error codes via API	Chapter 8
[SRS_BSW_00339] Reporting of production relevant errors and exceptions	Not Applicable
[SRS_BSW_00323] API parameter checking	Chapter 8
[SRS_BSW_00409] Header files for production code error IDs	Not Applicable
[SRS_BSW_00385] List possible error notifications	Chapter 7.3
[SRS_BSW_00386] Configuration for detecting an error	Chapter 7.4
[SRS_BSW_00415] User dependent include files	[SWS_BswM_00026
[SRS_BSW_00343] Specification and configuration of time	Chapter 10
[SRS_BSW_00346] Basic set of module files	Chapter 5
[SRS_BSW_00158] Separation of configuration from implementation	Chapter 5
[SRS_BSW_00370] Separation of callback interface from API	Chapter 8
[SRS_BSW_00357] Standard API return type	Chapter 8
[SRS_BSW_00377] Module specific API return types	Chapter 8
[SRS_BSW_00371] Do not pass function pointers via API	Chapter 8

[SRS_BSW_00358] Return type of init() functions	Chapter 8
[SRS_BSW_00414] Parameter of init function	Chapter 8
[SRS_BSW_00376] Return type and parameters of main processing functions	Chapter 8
[SRS_BSW_00359] Return type of callback functions	Chapter 8
[SRS_BSW_00360] Parameters of callback functions	Chapter 8
[SRS_BSW_00440] Function prototype for callback functions of AUTOSAR Services	Chapter 7.7
[SRS_BSW_00374] Module vendor identification	Chapter 10
[SRS_BSW_00379] Module identification	Chapter 10
[SRS_BSW_00003] Version identification	Chapter 10
[SRS_BSW_00318] Format of module version numbers	Chapter 10
[SRS_BSW_00321] Enumeration of module version numbers	Chapter 10

Document: AUTOSAR Requirements on Mode Management [4].

Requirement	***Satisfied by***
[SRS_ModeMgm_09177] Support of a configurable mode arbitration	SWS_BswM_00010, SWS_BswM_00012
[SRS_ModeMgm_09178] Support of Lists of Mode Dependant Actions	SWS_BswM_00016, SWS_BswM_00015
[SRS_ModeMgm_09175] Support of a configurable set of mode dependent enabled and concomitant disabled IPDU groups	SWS_BswM_00038
[SRS_ModeMgm_09176] Support of a configurable set of mode dependent to be enabled IPDU groups	SWS_BswM_00038
[SRS_ModeMgm_09183] Support of Mode dependent to be reset Signal Initial Values	SWS_BswM_00038
[SRS_ModeMgm_09174] Support of “Disable normal Communication”	SWS_BswM_00038
[SRS_ModeMgm_09179] Provision of an Interface to allow Mode Requests of SW C’s	Chapter 7.6 and SWS_BswM_00046
[SRS_ModeMgm_09180] Arbitration of Mode Requests	SWS_BswM_00009, SWS_BswM_00013, SWS_BswM_00014
[SRS_ModeMgm_09182] Local propagation of mode change information	SWS_BswM_00038

[SRS_ModeMgm_09184] Mode dependent activation and deactivation of IPDU groups	SWS_BswM_00038
[SRS_ModeMgm_09228] Provision of an Interface to allow Mode Requests of BSW Modules	SWS_BswM_00046, SWS_BswM_00047, SWS_BswM_00048, SWS_BswM_00049, SWS_BswM_00050, SWS_BswM_00051, SWS_BswM_00052
[SRS_ModeMgm_09229] Mode dependent callout	SWS_BswM_00039, SWS_BswM_00040
[SRS_ModeMgm_09230] All actions shall only be performed on mode change	SWS_BswM_00011, SWS_BswM_00023

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 and NAND. Each term in the expression corresponds to a mode request condition. These conditions verify if a requested or indicated mode is EQUAL or NOT_EQUAL to a certain mode. An example rule with two conditions is shown in Figure 2. The rules and the set of available logical operations are defined as a part of the ECU configuration described in chapter 10.2.

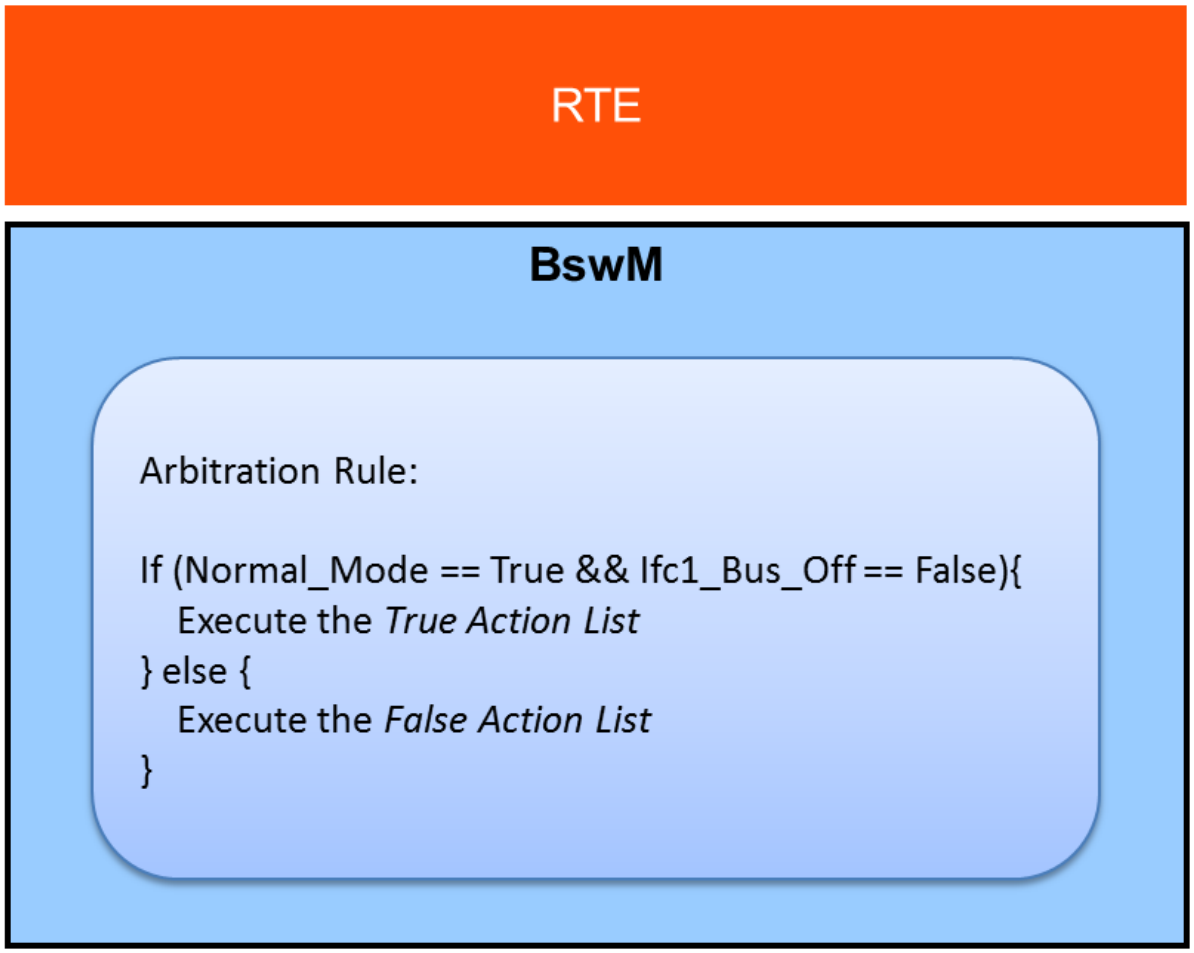


Figure 2: Pseudocode representation of an example rule with two conditions.

7.1.3 Requirements of Mode Arbitration

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

[SWS_BswM_00009] ⌈

The BswM shall perform mode arbitration based on incoming mode requests. ⌋ (SRS_ModeMgm_09180)

[SWS_BswM_00035] ⌈

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

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 ECUC_BswM_00856 configuration container.

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

[SWS_BswM_00012] ⌈
The mode arbitration rules shall be configurable using the module configuration parameters described in chapter 10.2.⌋(SRS_ModeMgm_09177)

[SWS_BswM_00117] ⌈
BswM is not allowed to use results from previous mode arbitrations as input for the logical expressions.⌋()

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

[SWS_BswM_00189] ⌈
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 environment of the caller. It is the responsibility of the system integrator to ensure that execution of the action list does not jeopardize system performance or consistency.
Especially, if the caller runs (or may run) in interrupt context, the restrictions concerning usage of system functions in interrupt context apply.

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

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

[SWS_BswM_00013] ⌈

It shall be possible to configure the BswM to execute the mode arbitration immediately upon a mode arbitration request. This is configured by setting the ECUC_BswM_00822 configuration parameter to BSWM_IMMEDIATE. 」(SRS_ModeMgm_09180)

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

[SWS_BswM_00014] 「
It shall (also) be possible to defer the mode arbitration until the execution of the main function of the BswM. This is configured by setting the ECUC_BswM_00822 configuration parameter to BSWM_DEFERRED.」(SRS_ModeMgm_09180)

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

[SWS_BswM_00068] 「
BswM shall delay mode arbitration requests received during the processing of its main function until it is finished.」()

[SWS_BswM_00069] 「
BswM shall delay mode arbitration requests received during the processing of an *immediate request* until it is finished.」()

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 ECUC_BswM_00805 in the configuration.

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

The initial value of each BswMModeRequestPort after initialization may be controlled by the configuration parameter BswMBswModeInitValue.

[SWS_BswM_00203]「
In case BswMModeInitValue is defined the BswM shall initialize the corresponding BswMModeRequestSource with the BswMBswModeInitValue value while the BswM is initialized. This 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 ([SRS_Rte_00116])

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 DEM event and/or cancel the currently executing action list.

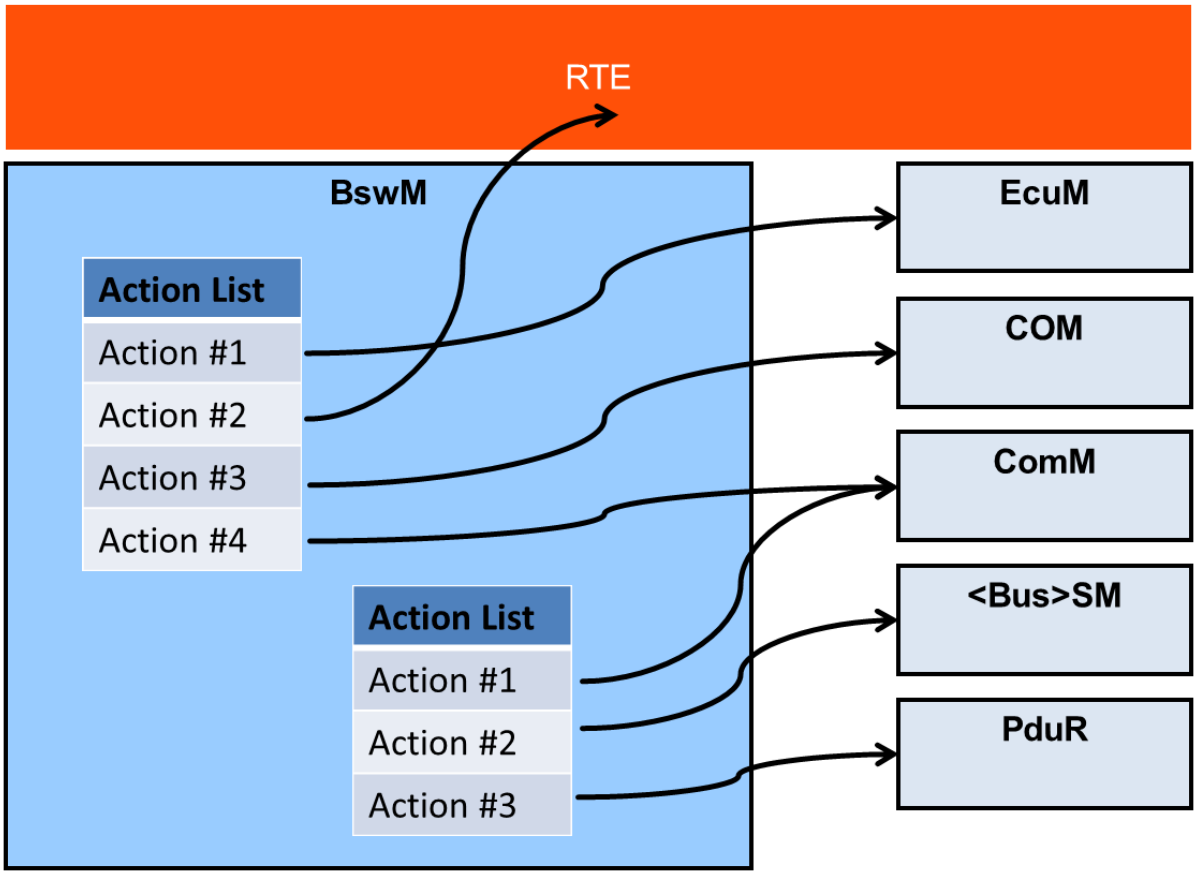


Figure 3: Example showing two action lists

As shown in Figure 3, 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 4 shows the minimal processing cycle for a Mode Request:

1. The Mode Requester SW-C requests mode A through its Sender Port. The RTE distributes the request and the BswM receives it through its Receiver Port.
2. The BswM evaluates its rules either as a result of a received mode arbitration request, or cyclically during the execution of the BswM main function.
3. 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 [10] as actions to inform the affected SW-Cs about the arbitration result. Any SW-C, especially the mode requester can register to receive the mode switch indications.

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

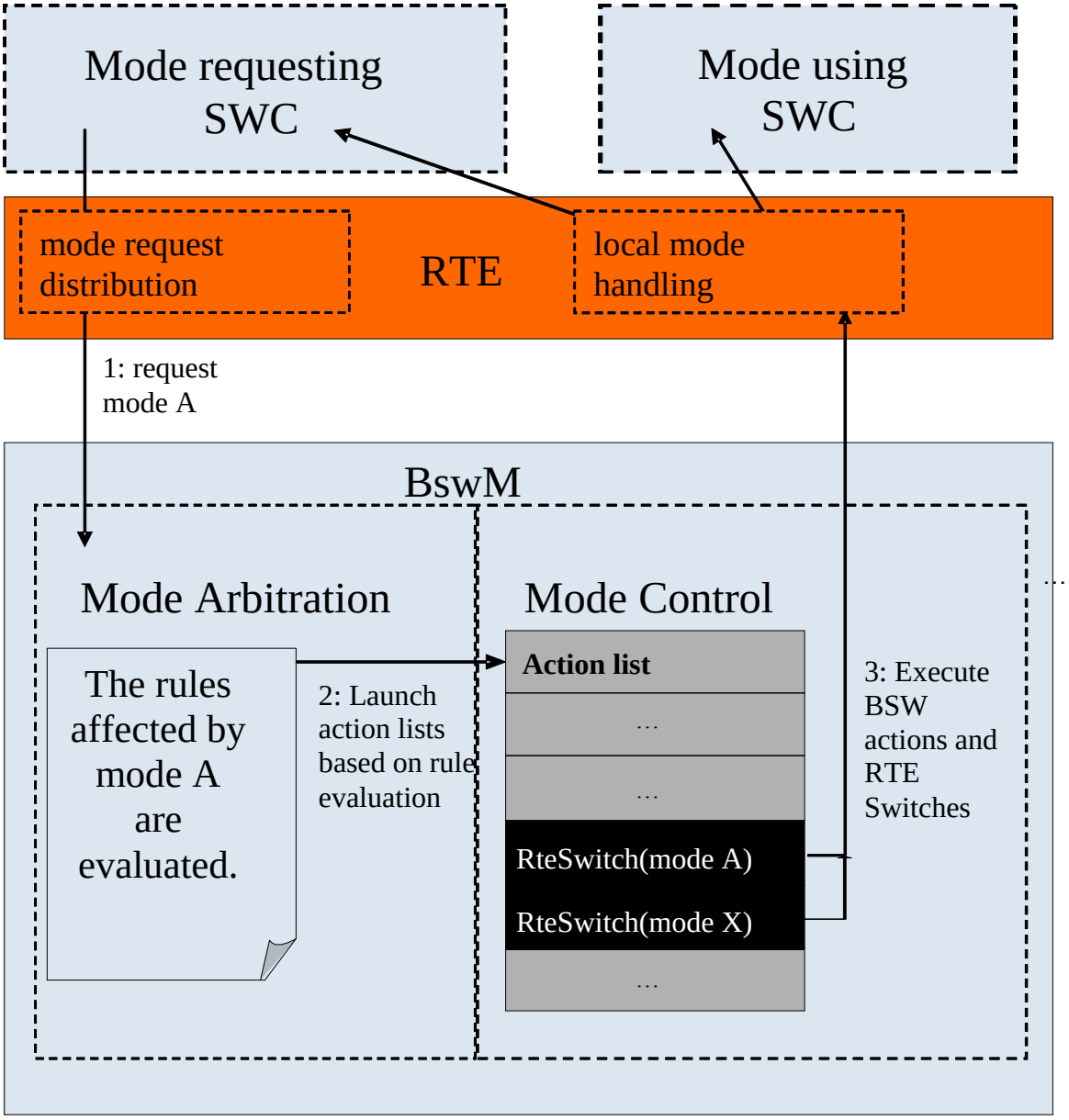


Figure 4: Mode Processing Cycle

7.2.2 Requirements on Mode Control

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

[SWS_BswM_00015] ⌈

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

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

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

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

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

[SWS_BswM_00037] ⌈
If cascaded action lists are used (i.e. using references to other rules or action lists) the action list structure may contain up to seven (7) hierarchic levels.
Note: The purpose of this limit is to make testing of BswM implementations and generator tools possible. The limit must be checked by the generator tool.⌋()

[SWS_BswM_00062] ⌈
Action lists associated with rules evaluated in the context of the mode arbitration request shall be executed by BswM immediately when triggered by the mode arbitration, and not be deferred to the main function execution.
Rationale: This allows very short latencies on mode requests when necessary. ⌋()

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.

[SWS_BswM_00011] ⌈

If a True action list is configured for triggered execution, the BswM shall only execute it when the evaluation of the corresponding rule changes from False to True.」(SRS_ModeMgm_09230)

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

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

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

[SWS_BswM_00055]「
The BswM shall abort the execution of an action list if an action returns E_NOT_OK and the corresponding BswMAbortOnFail configuration parameter is set to “true”.」()

[SWS_BswM_00121]「
The BswM shall report a DEM event if an action returns E_NOT_OK and the corresponding BswMReportFailToDemRef configuration parameter is set.」()

7.2.4 Available Actions

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

[SWS_BswM_00038]「

BswM shall be able to execute the predefined actions defined by BswM0826_Conf.」(SRS_ModeMgm_09175, BSW009176, SRS_ModeMgm_09183,SRS_ModeMgm_09174, SRS_ModeMgm_09182, SRS_ModeMgm_09184)

[SWS_BswM_00039]「
The BswM shall be able to call any function in the AUTOSAR BSW even though it is not among the standardized actions defined in [SWS_BswM_00038. 」(SRS_ModeMgm_09229)

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

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

[SWS_BswM_00190] ⌈
The BswM shall only execute actions which are available on its own partition.⌋()

7.2.5 Behavior of Mode Control after Initialization

The behavior of the Mode Control after initialization of the BswM is configured by the ECUC_BswM_00888 parameter. It defines the “previous evaluation result” to be used when deciding on what action list to execute after the first evaluation of a rule after initialization. The configuration parameter BswMActionListExecution also affects the action list execution after initialization.

[SWS_BswM_00066] ⌈
The BswM shall act according to what is stated in Table 2 when a rule is evaluated for the first time after initialization.⌋()

BswMRuleInitState	***BswMActionListExecution***	***Rule evaluated to true***	***Rule evaluated to false***
BSWM_UNDEFINED	BSWM_TRIGGER	Execute "true” action list	Execute “false” action list
BSWM_TRUE	BSWM_TRIGGER	Do nothing	Execute “false” action list
BSWM_FALSE	BSWM_TRIGGER	Execute “true” action list	Do nothing
BSWM_UNDEFINED	BSWM_CONDITION	Execute "true” action list	Execute “false” action list
BSWM_TRUE	BSWM_CONDITION	Execute "true” action list	Execute “false” action list
BSWM_FALSE	BSWM_CONDITION	Execute “true” action list	Execute “false” action list

Table 2: Usage of the BswMRuleInitState configuration parameter

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

7.2.6 Handling of I-PDU group switching

BswM is the only module that controls the starting and stopping of I-PDU groups. As such, Com does not provide accessor functions for the I-PDU group states, so the BswM must maintain an internal Com_IpduGroupVector to implement this action. Also, to perform I-PDU group switches (enable/disable) in an efficient and consistent way, the BswM shall perform the actual I-PDU Group Control function call at the end of processing the main function or an immediate processing request. Essentially, this means that the I-PDU Group Switch action manipulates an I-PDU Group Vector and a ReInit flag that is internal to the BswM and that these internal variables are passed as parameters to COM.

[SWS_BswM_00128] ⌈
BswM shall keep internal variables as an accumulative storage of the results of BswMPduGroupSwitch actions. These internal variables shall be initialized to all-zeros when the BswM is initialized. These internal variables shall be used as the parameters when calling the Com_IpduGroupControl() function.⌋()

[SWS_BswM_00129] ⌈
If any BswMPduGroupSwitch action(s) have been performed, the BswM shall execute the Com_IpduGroupControl command at the end of its processing of the BswM main function or an immediate request processing.⌋()

7.3 Multi Partition Support

The BswM mainly interacts with the state managers of the functional clusters, e.g. with the ComM, and should therefore be locally available on the same cores if inter-core communication should be limited as much as possible.
Therefore, the BswM is distributed over multiple partitions that contain BSW modules. These independent BswMs have partition specific configuration sets. The synchronization of the different instances is done via BSW Modes.

[SWS_BswM_00191] ⌈
If a partition of the ECU contains a BSW module that is running inside this partition, this partition also should have a BswM.⌋()
The BswM is responsible for the complete initialization of the BSW Modules in a specific partition. As the initialization sequence largely depends on the distribution of the modules in different partitions it has a big impact on the configuration of the BswM.

[SWS_BswM_00192] ⌈

Each BswM should coordinate the initialization of the BSW modules which are running in the same partition.」()

Each instance of the BswM will then take care of the correct initialization and deinitialization of the partition local BSW modules, so that the following scenarios can be realized:

Startup up:
After startup of the OS, each EcuM will hand over control to the partition local BswM, which then takes care of the initialization of the other partition local BSW Modules. Afterwards, it signals the readiness of the partition to the other BswM instances.

Shutdown:
The partition local BswM determines via its ModeRequestSources, whether it can be shut down or not. If this is the case, it signals its current state to the other BswM instances. The BswM placed inside the partition of the Master EcuM can then decide on this information whether it initiates a shutdown of the ECU.

Deinitialization:
The BswM (on the partition where the Master EcuM is running inside) can signal the other BswMs that it wants to shutdown the ECU. The BwsM must then deinitialize the modules running inside their partition to enable a clean shutdown.

Restart of a partition:
If a partition is restarted, the local BswM signals to the other instances that it is in a restart mode. Then, they have to determine if local applications need to be informed or potentially restarted, and how to synchronize them to the newly started partition.

7.4 Debugging Support

For details refer to the chapter “Debugging support” in *SWS_BSWGeneral.*

7.5 Error classification

Type or error	*Relevance*	*Related error code*	*Value [hex]*
A service was called prior to initialization	Development	BSWM_E_NO_INIT	0x01
A null pointer was passed as an argument	Development	BSWM_E_NULL_POINTER	0x02
A parameter was invalid (unspecific)	Development	BSWM_E_PARAM_INVALID	0x03
A requesting user was out of range	Development	BSWM_E_REQ_USER_OUT_OF_RANGE	0x04
A requested mode was out of range	Development	BSWM_E_REQ_MODE_OUT_OF_RANGE	0x05

The provided configuration is inconsistent	Development	BSWM_E_PARAM_CONFIG	0x06
A parameter pointer was invalid	Development	BSWM_E_PARAM_POINTER	0x07
An action returned E_NOT_OK	Production	BSWM_E_ACTION_FAILED	

Table 3: Table of development errors

7.6 Error detection

For details refer to the relevant chapter in *SWS_BSWGeneral.*

7.7 Error notification

For details refer to the relevant chapter in *SWS_BSWGeneral.*

7.8 Version checking

For details refer to the relevant chapter in *SWS_BSWGeneral.*

7.9 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 7.9.1. 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 7.9.3 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 0 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 5 shows how the ports of the application SW-Cs connect to the service ports of the BSW Mode Manager. The Application Mode Manager SW-C has a Mode Request Port and a Mode Switch R-port (named `modeNotificationPort` to distinguish it from the Mode Switch P-ports). The first port is to request changes in its application mode, the latter to receive notifications when the BswM has performed the mode change. The Mode Request Port of the Application Mode Manager (`modeRequestPort0`) connects to the corresponding Mode Request Port of the BSW Mode Manager. Since this is normal Sender/Receiver communication, the Application Mode Manager may connect to multiple BSW Mode Managers even on remote ECUs.

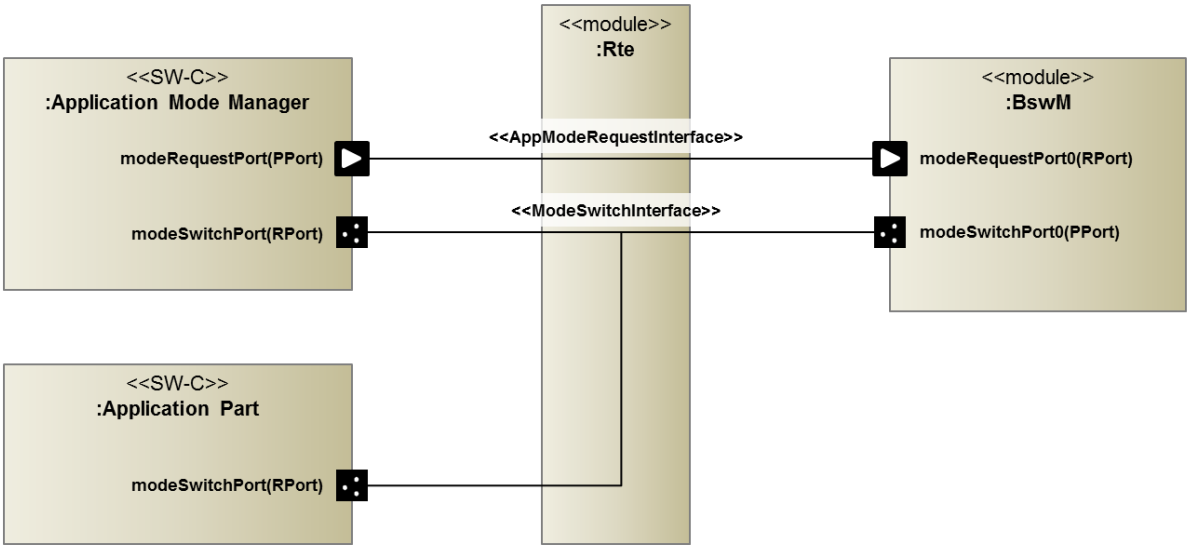


Figure 5: Connections between Application Mode Manager, Application Parts and the BSW Mode Manager

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.

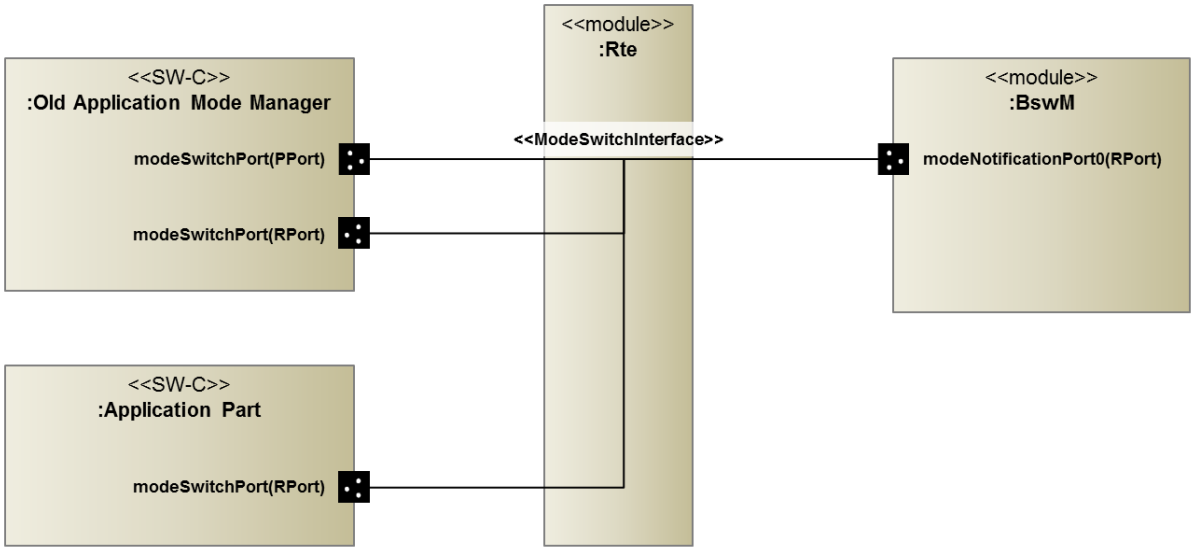


Figure 6: Connections between SW-C based Application Mode Manager, Application Parts and the BSW Mode Manager

Figure 6 shows that SW-C based Application Mode Managers (as used in AUTOSAR R3.1 and earlier) switch the application mode directly and do not request it from the BSW Mode Manager. Therefore, they directly connect a Mode Switch Port to the

local RTE. This implies that the application mode needs to be local to that ECU and that no arbitration in the BSW Mode Manager is possible. Nevertheless, the BSW Mode Manager may use the current application mode as an input for its rules because it can have a Mode Switch R-Port (named `modeNotificationPort0` in the figure) for this application mode.

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

7.9.1 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 modeRequestPort_{Name};
```

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

```
Rte_Read_modeRequestPort_{Name}_requestedMode( &<variable> );
```

[SWS_BswM_00220]⌈BswMSwcModeRequests contain a foreign reference to the ModeDeclarationGroupPrototype used by the incoming mode request. As the mode request is SR-Communication the BswM shall provide a SR-Interface which corresponds to the ModeDeclarationGroupPrototype. The SR-Interface shall contain one ApplicationPrimitiveDataType which is defined as an enumeration (compuMethod) with the enumeration literals matching the ModeDeclarations.⌋()

7.9.2 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_{Name};
```

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_{Name}_currentMode( <new_mode> );

SchM_Switch_modeSwitchPort_{Name}_currentMode( <new_mode> );
```

[SWS_BswM_00221] 「 BswMSwitchPort specifies PPorts and/or providedModeDeclarationGroups, which the BswM shall create in its SWCD resp. BSWMD. If the container is referenced by one or more BswMRteSwitchActions the BswM shall create a corresponding PPort in its Service Description. If the container is referenced by a BswMSchMSwitch action the BswM shall create the corresponding ModeDeaclarationGroupPrototype as providedModeDeclarationGroup in it BSWMD.
If the container is referenced by BswMSchMSwitch AND BswmRteSwitchActions, the providedModeDeclarationGroup and a PPort shall be created. In the corresponding SwcBswMapping a synchronizedModeGroup has to be created. See also chapter 4.4.7 in SWS_RTE.」()

7.9.3 Notifications of Mode Switches

In addition to mode requests, the currently active modes can also be used 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_{Name};
```

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

```
Rte_Mode_modeNotificationPort_{Name}_currentMode( &<variable> );

SchM_Mode_modeNotificationPort_{Name}_currentMode( &<variable> );
```

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

```
Rte_Mode_modeNotificationPort_{Name}_currentMode( &<variable>,
&<previousmode>, &<nextmode> );

SchM_Mode_modeNotificationPort_{Name}_currentMode( &<variable>,
&<previosmode>, &<nextmode> );
```

[SWS_BswM_00222]「 BswMBswModeNotification is used to configure mode requests from a BSW Module. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_IMMEDIATE then it is assumed that the Basic Software Module Description of the BswM contains a BswSchedulableEntity which is activated by a BswModeSwitchEvent. This BswModeSwitchEvent shall refer to the ModeDeclarationGroupPrototype which is referenced by BswMBswModeDeclarationGroupPrototypeRef. If the BswMRequestProcessing of this BswMModeRequestPort is set to BSWM_DEFERRED then it is up to the implementer of the BswM tooling whether a BswSchedulableEntity is used to update the BswM internal mode mirror or whether the BswM internal mode mirror is updated during the main function execution.」()

7.9.4 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:

[SWS_BswM_00138]「

```
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 ServiceProxyComponentType instead of the normal ServiceComponentType.

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

7.9.5 BswM Service

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

[SWS_BswM_00137] ⌈

```
ServiceComponentType BswM {

  // For each mode request
  RequirePort <request_interface> modeRequestPort_{Name};

  // For each mode switch
  ProvidePort <mode_interface> modeSwitchPort_{Name};

  // For each mode notification
  RequirePort <mode_interface> modeNotificationPort_{Name};

  InternalBehavior {

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

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

  };

};
```

⌋()

7.10 Pretended Networking

The current version of the BswM SWS supports Pretended Networking only for the Can bus through the API BswM_CanSM_CurrentIcomConfiguration and the configuration container BswMCanSMIcomIndication.

The AUTOSAR_EXP_ModemanagementGuide document contains guidelines for the BswM configuration regarding Pretended Networking.

8 API specification

8.1.1 Imported types

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

[SWS_BswM_00001] ⌈The BSW Mode Manager shall use only the following imported types of other modules:

Module	*Imported Type*
CanSM	CanSM_BswMCurrentStateType
Com	Com_IpduGroupIdType
	Com_IpduGroupVector
ComM	ComM_InhibitionStatusType
	ComM_InitStatusType
	ComM_ModeType
	ComM_PncModeType
	ComM_UserHandleType
ComStack_Types	IcomConfigIdType
	IcomSwitch_ErrorType
	NetworkHandleType
	PNCHandleType
	PduIdType
Dcm	Dcm_CommunicationModeType
Dem	Dem_EventIdType
	Dem_EventStatusType
EcuM	EcuM_StateType
	EcuM_WakeupSourceType
	EcuM_WakeupStatusType
EthSM	EthSM_NetworkModeStateType
FrSm	FrSM_BswM_StateType
J1939Dcm	J1939Dcm_StateType
J1939Rm	J1939Rm_StateType
LinIf	LinIf_SchHandleType
	LinTp_Mode
LinSM	LinSM_ModeType
McOs	CoreIdType
Nm	Nm_StateType
NvM	NvM_BlockIdType
	NvM_RequestResultType
Os	ApplicationType
	IdleModeType
	StatusType
Sd	Sd_ClientServiceCurrentStateType
	Sd_ConsumedEventGroupCurrentStateType
	Sd_EventHandlerCurrentStateType
	Sd_StateType
Std_Types	Std_ReturnType
	Std_VersionInfoType

」()

8.1.2 Type definitions

[SWS_BswM_00041] 「The following Data Types shall be used for the functions defined in this specification.」()

8.1.2.1 BswM_ConfigType

[SWS_BswM_00213]「

Name:	BswM_ConfigType	
Type:	Structure	
Range:	-	The contents of this structure depends on the configuration variant.
Description:	This structure contains all post-build configurable parameters of the BSW Mode Manager. A pointer to this structure is passed to the BSW Mode Manager initialization function for configuration.	

」()

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

8.1.2.2 BswM_ModeType

[SWS_BswM_00214]「

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

」()

8.1.2.3 BswM_ModeGroupType

[SWS_BswM_00215]「

Name:	BswM_ModeGroupType		
Type:	uint8, uint16		
Range:	0-255, 0-65535	--	The range of valid IDs depends on configuration and on the chosen platform type.
Description:	This type identifies the mode group that can be managed by a BswM.		

」()

8.1.2.4 BswM_UserType

[SWS_BswM_00216]「

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

」()

8.1.3 Function definitions

8.1.3.1 BswM_BswMModeRequest

[SWS_BswM_00187] ⌜

Service name:	BswM_BswMModeRequest	
Syntax:	`void BswM_BswMModeRequest( BswM_ModeGroupType requested_modeGroup, BswM_ModeType requested_mode )`	
Service ID[hex]:	0x1c	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	requested_modeGroup	The mode group that the mode was requested for
	requested_mode	The requested mode
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by a BswM on another partition to request a mode.	

⌟()

The corresponding configuration container for this API is BswMBswMModeRequest.

8.1.3.2 BswM_BswMModeSwitchNotification

[SWS_BswM_00188] ⌜

Service name:	BswM_BswMModeSwitchNotification	
Syntax:	`void BswM_BswMModeSwitchNotification( BswM_ModeGroupType switched_modeGroup, BswM_ModeType switched_mode )`	
Service ID[hex]:	0x1d	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	switched_modeGroup	The mode group in that the mode was switched
	switched_mode	The mode that was switched to.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called when a BswM on a remote partition has switched a mode.	

⌟()

The corresponding configuration container for this API is BswMBswMModeSwitchNotification.

8.1.3.3 BswM_BswMPartitionRestarted

[SWS_BswM_00193] ⌜

Service name:	BswM_BswMPartitionRestarted

Syntax:	`void BswM_BswMPartitionRestarted(`
	`  void`
	`)`

Service ID[hex]:	0x1e
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Function called by Restart Task if the partition containing the BswM has been restarted.

」()

The corresponding configuration container for this API is BswMPartitionRestarted.

8.1.3.4 BswM_CanSM_CurrentIcomConfiguration

[SWS_BswM_00164] 「

Service name:	BswM_CanSM_CurrentIcomConfiguration	

Syntax:	`void BswM_CanSM_CurrentIcomConfiguration(`
	`  NetworkHandleType Network,`
	`  IcomConfigIdType ActiveConfiguration,`
	`  IcomSwitch_ErrorType Error`
	`)`	

Service ID[hex]:	0x1a	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Network	The CAN channel the requested state corresponds to.
	ActiveConfiguration	The configuration Id of the Icom configuration.
	Error	ICOM_SWITCH_E_OK: No Error ICOM_SWITCH_E_FAILED: Switch to requested Configuration failed. Severe Error.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function to inform BswM about the switch of Icom Configuration.	

」()

The corresponding configuration container for this API is BswMCanSMIcomIndication.

[SWS_BswM_00166] 「
The interface BswM_CanSM_CurrentIcomConfiguration shall be used by the CanSM to inform the BswM about the activation of a new ICOM configuration for a given channel.」()

8.1.3.5 BswM_CanSM_CurrentState

[SWS_BswM_00049] ⌈

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.	

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMCanSMIndication.

[SWS_BswM_00080] ⌈
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the state indication and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(SRS_BSW_00406)

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

8.1.3.6 BswM_ComM_CurrentMode

[SWS_BswM_00047] ⌈

Service name:	BswM_ComM_CurrentMode	
Syntax:	`void BswM_ComM_CurrentMode( NetworkHandleType Network, ComM_ModeType RequestedMode )`	
Service ID[hex]:	0x0e	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Network	The ComM communication channel that the indicated state corresponds to.
	RequestedMode	The current state of the ComM communication channel
Parameters (inout):	None	

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

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMComMIndication.

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

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

8.1.3.7 BswM_ComM_CurrentPNCMode

[SWS_BswM_00148] ⌈

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

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMComMPncRequest.

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

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

8.1.3.8 BswM_ComM_InitiateReset

[SWS_BswM_00217] ⌈

Service name:	BswM_ComM_InitiateReset

Syntax:	`void BswM_ComM_InitiateReset(`
	`  void`
	`)`

Service ID[hex]:	0x22
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Function called by ComM to signal a shutdown.

⌋()

The corresponding configuration container for this API is BswMComMInitiateReset.

8.1.3.9 BswM_Dcm_ApplicationUpdated

[SWS_BswM_00158] ⌈

Service name:	BswM_Dcm_ApplicationUpdated

Syntax:	`void BswM_Dcm_ApplicationUpdated(`
	`  void`
	`)`

Service ID[hex]:	0x14
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This function is called by the DCM in order to report an updated application.

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMDcmApplicationUpdatedIndication.

[SWS_BswM_00159:] ⌈

If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(SRS_BSW_00406)

8.1.3.10 BswM_Dcm_CommunicationMode_CurrentState

[SWS_BswM_00048] ⌈

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

⌋(SRS_ModeMgm_09228)

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.

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

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

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.

8.1.3.11 BswM_Deinit

[SWS_BswM_00119] ⌈

Service name:	BswM_Deinit

Syntax:	`void BswM_Deinit(`
	`  void`
	`)`

Service ID[hex]:	0x04
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Deinitializes the BSW Mode Manager.

⌋()

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

8.1.3.12 BswM_EcuM_CurrentState

[SWS_BswM_00056] ⌈

Service name:	BswM_EcuM_CurrentState	

Syntax:	`void BswM_EcuM_CurrentState(`
	`  EcuM_StateType CurrentState`
	`)`	

Service ID[hex]:	0x0f	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	CurrentState	The requested ECU Operation Mode
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by EcuM to indicate the current ECU Operation Mode.	

⌋()

The corresponding configuration container for this API is BswMEcuMIndication.

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

[SWS_BswM_00103] ⌈

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

8.1.3.13 BswM_EcuM_CurrentWakeup

[SWS_BswM_00131] ⌈

Service name:	BswM_EcuM_CurrentWakeup	

Syntax:	`void BswM_EcuM_CurrentWakeup(`
	`  EcuM_WakeupSourceType source,`
	`  EcuM_WakeupStatusType state`
	`)`	

Service ID[hex]:	0x10	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	source	Wakeup source(s) that changed state.
	state	The new state of the wakeup source(s)
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by EcuM to indicate the current state of a wakeup source.	

⌋()

The corresponding configuration container for this API is BswMEcuMWakeupSource.

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

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

8.1.3.14 BswM_EthSM_CurrentState

[SWS_BswM_00050] ⌈

Service name:	BswM_EthSM_CurrentState

Syntax:	`void BswM_EthSM_CurrentState(`
	`  NetworkHandleType Network,`
	`  EthSM_NetworkModeStateType CurrentState`
	`)`

Service ID[hex]:	0x0d
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	Network	The Ethernet channel that the indicated state corresponds to.
	CurrentState	The current state of the Ethernet channel.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by EthSM to indicate its current state.	

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMEthSMIndication.

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

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

8.1.3.15 BswM_FrSM_CurrentState

[SWS_BswM_00051] ⌈

Service name:	BswM_FrSM_CurrentState	
Syntax:	`void BswM_FrSM_CurrentState( NetworkHandleType Network, FrSM_BswM_StateType CurrentState )`	
Service ID[hex]:	0x0c	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Network	The FlexRay cluster that the indicated state corresponds to.
	CurrentState	The corrent state of the FlexRay cluster.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by FrSM to indicate its current state.	

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMFrSMIndication.

[SWS_BswM_00082] ⌈
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the state

indication and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

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

8.1.3.16 BswM_GetVersionInfo

[SWS_BswM_00003]「

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

」(SRS_BSW_00407)

[SWS_BswM_00139]「
If the BswMDevErrorDetect switch is enabled, the function BswM_GetVersionInfo shall check if a NULL pointer is passed for the VersionInfo parameter.
In case of an error the function BswM_GetVersionInfo shall not be executed and an error BSWM_E_PARAM_POINTER shall be reported to the Development Error Tracer.」()

8.1.3.17 BswM_Init

[SWS_BswM_00002]「

Service name:	BswM_Init	
Syntax:	`void BswM_Init( const BswM_ConfigType * ConfigPtr )`	
Service ID[hex]:	0x00	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	ConfigPtr	Pointer to post-build configuration data
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	

Description:	Initializes the BSW Mode Manager.

」(SRS_BSW_00101)

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

[SWS_BswM_00044] 「This routine shall initialize all module global variables of the BSW Mode Manager.」()

[SWS_BswM_00118] 「BswM_Init shall only require the OS and the SchM to be initialized before it can be called.」()

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

[SWS_BswM_00088] 「
If the BswMDevErrorDetect switch is enabled and the configuration variant is VARIANT-POST-BUILD, the function BswM_Init shall check if a NULL pointer is passed for the ConfigPtr parameter. In case of an error the remaining function BswM_Init shall not be executed and the function BswM_Init shall report development error code BSWM_E_NULL_POINTER to the Det_ReportError service of the Development Error Tracer.」()

8.1.3.18 BswM_J1939DcmBroadcastStatus

[SWS_BswM_00165] 「

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. 1: Network enabled 0: Network disabled
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.	

」()

The corresponding configuration container for this API is BswMJ1939DcmBroadcastStatus.

8.1.3.19 BswM_J1939Nm_StateChangeNotification

[SWS_BswM_00194] ⌈

Service name:	BswM_J1939Nm_StateChangeNotification	

Syntax:	`void BswM_J1939Nm_StateChangeNotification(`
	`  NetworkHandleType Network,`
	`  uint8 Node,`
	`  Nm_StateType NmState`
	`)`	

Service ID[hex]:	0x18	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Network	Identification of the J1939 channel
	Node	Identification of the J1939 node
	NmState	Current (new) state of the J1939 node
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Notification of current J1939Nm state after state changes.	

⌋()

The corresponding configuration container for this API is BswMJ1939NmIndication.

8.1.3.20 BswM_LinSM_CurrentSchedule

[SWS_BswM_00058] ⌈

Service name:	BswM_LinSM_CurrentSchedule	

Syntax:	`void BswM_LinSM_CurrentSchedule(`
	`  NetworkHandleType Network,`
	`  LinIf_SchHandleType CurrentSchedule`
	`)`	

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

⌋()

The corresponding configuration container for this API is BswMLinScheduleIndication.

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

8.1.3.21 BswM_LinSM_CurrentState

[SWS_BswM_00052] ⌈

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

⌋(SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMLinSMIndication.

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

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

8.1.3.22 BswM_LinTp_RequestMode

[SWS_BswM_00156] ⌈

Service name:	BswM_LinTp_RequestMode
Syntax:	`void BswM_LinTp_RequestMode( NetworkHandleType Network, LinTp_Mode LinTpRequestedMode`

	)	
Service ID[hex]:	0x0b	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Network	The LIN channel that the LinTp mode request relates to.
	LinTpRequestedMode	The requested LIN TP mode.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by LinTP to request a mode for the corresponding LIN channel. The LinTp_Mode correlates to the LIN schedule table that should be used.	

」()

The corresponding configuration container for this API is BswMLinTpModeRequest.

[SWS_BswM_00112] 「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

[SWS_BswM_00113] 「
If the BswMDevErrorDetect switch is enabled, the parameter LinTpRequestedMode shall be checked for being in the allowed range. In case of an error the BswM shall ignore the mode request and report the error, to the Development Error Tracer with the value BSWM_E_REQ_MODE_OUT_OF_RANGE.」()

8.1.3.23 BswM_NvM_CurrentBlockMode

[SWS_BswM_00104] 「

Service name:	BswM_NvM_CurrentBlockMode	

Syntax:	`void BswM_NvM_CurrentBlockMode(`
	`  NvM_BlockIdType Block,`
	`  NvM_RequestResultType CurrentBlockMode`
	`)`	

Service ID[hex]:	0x16	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	Block	The Block that the new NvM Mode corresponds to.
	CurrentBlockMode	The current block mode of the NvM block.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by NvM to indicate the current block mode of an NvM block. To use this function integration code will be needed.	

」()

The corresponding configuration container for this API is BswMNvMRequest.

[SWS_BswM_00109] 「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the block mode indication and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

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

8.1.3.24 BswM_NvM_CurrentJobMode

[SWS_BswM_00152] 「

Service name:	BswM_NvM_CurrentJobMode	
Syntax:	`void BswM_NvM_CurrentJobMode( uint8 ServiceId, NvM_RequestResultType CurrentJobMode )`	
Service ID[hex]:	0x17	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ServiceId	Indicates whether the callback refers to multi block services NvM_ReadAll or NvM_WriteAll.
	CurrentJobMode	Current state of the multi block job indicated by parameter ServiceId.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by NvM to inform the BswM about the current state of a multi block job.	

」()

The corresponding configuration container for this API is BswMNvMJobModeIndication.

[SWS_BswM_00153] 「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the job mode indication and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

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

8.1.3.25 BswM_RequestMode

[SWS_BswM_00046] ⌈

Service name:	BswM_RequestMode	

Syntax:	`void BswM_RequestMode(`
	`  BswM_UserType requesting_user,`
	`  BswM_ModeType requested_mode`
	`)`	

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

⌋(SRS_ModeMgm_09179, SRS_ModeMgm_09228)

The corresponding configuration container for this API is BswMGenericRequest.

[SWS_BswM_00077] ⌈
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.⌋(SRS_BSW_00406)

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

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

8.1.3.26 BswM_Sd_ClientServiceCurrentState

[SWS_BswM_00204] ⌈

Service name:	BswM_Sd_ClientServiceCurrentState	
Syntax:	`void BswM_Sd_ClientServiceCurrentState( uint16 SdClientServiceHandleId, Sd_ClientServiceCurrentStateType CurrentClientState )`	
Service ID[hex]:	0x1f	
Sync/Async:	Asynchronous	
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).	

⌋()

The corresponding configuration container for this API is BswMSdClientServiceCurrentState.

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

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

8.1.3.27 BswM_Sd_ConsumedEventGroupCurrentState

[SWS_BswM_00207] ⌈

Service name:	BswM_Sd_ConsumedEventGroupCurrentState
Syntax:	`void BswM_Sd_ConsumedEventGroupCurrentState( uint16 SdConsumedEventGroupHandleId, Sd_ConsumedEventGroupCurrentStateType ConsumedEventGroupState )`
Service ID[hex]:	0x21
Sync/Async:	Asynchronous
Reentrancy:	Reentrant

Parameters (in):	SdConsumedEventGroupHandleId	HandleId to identify the Consumed Eventgroup
	ConsumedEventGroupState	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).	

⌋()

The corresponding configuration container for this API is BswMSdConsumedEventGroupCurrentState.

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

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

8.1.3.28 BswM_Sd_CurrentState

[SWS_BswM_00160] ⌈

Service name:	BswM_Sd_CurrentState	

Syntax:	`void BswM_Sd_CurrentState(`
	`  uint16 ServiceInstanceID,`
	`  Sd_StateType CurrentState`
	`)`	

Service ID[hex]:	0x19	
Sync/Async:	Asynchronous	
Reentrancy:	Reentrant	
Parameters (in):	ServiceInstanceID	ID to identify the Client Service Instance.
	CurrentState	Current state of the Server Service Instance.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Function called by Sd to indicate current state of Sd.	

⌋()

[SWS_BswM_00161] ⌈

If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

8.1.3.29 BswM_Sd_EventHandlerCurrentState

[SWS_BswM_00210] 「

Service name:	BswM_Sd_EventHandlerCurrentState	
Syntax:	`void BswM_Sd_EventHandlerCurrentState( uint16 SdEventHandlerHandleId, Sd_EventHandlerCurrentStateType EventHandlerStatus )`	
Service ID[hex]:	0x20	
Sync/Async:	Asynchronous	
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).	

」()

The corresponding configuration container for this API is BswMSdEventHandlerCurrentState.

[SWS_BswM_00211] 「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

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

8.1.3.30 BswM_TriggerSlaveRTEStop

[SWS_BswM_00141] 「

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

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

」()

The corresponding configuration container for this API is BswMTriggerSlaveRTEStop.

[SWS_BswM_00143] 「
This function BswM_TriggerSlaveRTEStop starts a OS task on the slave core which stops the RTE.」()

[SWS_BswM_00144] 「
The function BswM_TriggerSlaveRTEStop shall return the result of the OS task creation.」()

8.1.3.31 BswM_TriggerStartUpPhase2

[SWS_BswM_00140] 「

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

」()

The corresponding configuration container for this API is BswMTriggerStartUpPhase2.

[SWS_BswM_00142] 「

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

[SWS_BswM_00145]「
The function BswM_TriggerStartUpPhase2 shall return the result of the OS task creation.」()

8.1.3.32 BswM_WdgM_RequestPartitionReset

[SWS_BswM_00157]「

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

」()

The corresponding configuration container for this API is BswMWdgMRequestPartitionReset.

[SWS_BswM_00134]「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. In case of an error, the BswM shall ignore the mode request and report the error to the Development Error Tracer with the error code BSWM_E_NO_INIT.」(SRS_BSW_00406)

Usage/Configuration clarification:
1. The BswMModeRequestPort's BswMRequestProcessing parameter must be configured as BSWM_IMMEDIATE.
2. The BswMModeRequestPort must not have a BswMInitialValue container configured.
3. The BswMModeCondition referencing the BswMModeRequestPort must not have BswMConditionValue container configured.
4. The BswMRule's BswMLogicalExpression must only reference the BswMModeCondition that references this BswMModeRequestPort. If further conditions need to be evaluated, then another BswMRule may be added to the BswMRule's TrueActionList.
5. The BswMRule associated with that request must not have a FalseActionList.
6. The BswMActionList referenced as TrueActionList can only have the BswMActionListExecution parameter configured as BSWM_CONDITION.

8.1.4 Expected Interfaces

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

8.1.4.1 Mandatory Interfaces

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

[SWS_BswM_00007] ⌈

API function	*Description*
Dem_ReportErrorStatus	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation.

⌋()

8.1.4.2 Optional Interfaces

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

[SWS_BswM_00008] ⌈

API function	*Description*
ComM_GetCurrentComMode	Function to query the current Communication Mode. ComM shall use the corresponding interfaces of the Bus State Managers to get the current Communication Mode of the network. (Call to Bus State Manager API: XXXSM _GetCurrentComMode(...))
ComM_GetInhibitionStatus	Returns the inhibition status of a ComM channel.
ComM_GetMaxComMode	Function to query the maximum allowed Communication Mode of the corresponding user.
ComM_GetRequestedComMode	Function to query the currently requested Communication Mode of the corresponding user.
ComM_GetStatus	Returns the initialization status of the AUTOSAR Communication Manager. After a call to ComM_DeInit() ComM should have status COMM_UNINIT, and a new call to ComM_Init needed to make sure ComM restart internal state machines to defailt values.
ComM_GetVersionInfo	This function returns the published information (for details refer to table 10.3)
ComM_LimitChannelToNoComMode	Changes the inhibition status for the channel for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_LimitECUToNoComMode, same functionality but for all channels)
ComM_LimitECUToNoComMode	Changes the inhibition status for the ECU (=all channels) for changing from COMM_NO_COMMUNICATION to a higher Communication Mode. (See also ComM_LimitChannelToNoComMode, same functionality but for a specific channels)
ComM_PreventWakeUp	Changes the inhibition status COMM_NO_WAKEUP for the corresponding channel.

ComM_ReadInhibitCounter	This function returns the amount of rejected COMM_FULL_COMMUNICATION user requests.
ComM_RequestComMode	Requesting of a Communication Mode by a user. Note: Internally mode COMM_SILENT_COMMUNICATION is not a valid request for a user, mode used for synchronization at shutdown. Valid modes are COMM_NO_COMMUNICATION and COMM_FULL_COMMUNICATION
ComM_ResetInhibitCounter	This function resets the Inhibited COMM_FULL_COMMUNICATION request Counter.
ComM_SetECUGroupClassification	Changes the ECU Group Classification status (see chapter 10.2.2)
Com_ClearIpduGroupVector	This service sets all bits of the given Com_IpduGroupVector to 0.
Com_IpduGroupControl	This service starts I-PDU groups.
Com_ReceptionDMControl	This service enables or disables I-PDU group Deadline Monitoring.
Com_SetIpduGroup	This service sets the value of a bit in an I-PDU group vector.
Com_SwitchIpduTxMode	The service Com_SwitchIpduTxMode sets the transmission mode of the I-PDU referenced by PduId to Mode. In case the transmission mode changes, the new mode shall immediately be effective (see SWS_Com_00239). In case the requested transmission mode was already active for this I-PDU, the call will have no effect.
ControlIdle	This API allows the caller to select the idle mode action which is performed during idle time of the OS (e.g. if no Task/ISR is active). It can be used to implement energy savings. The real idle modes are hardware dependent and not standardized. The default idle mode on each core is IDLE_NO_HALT.
Det_ReportError	Service to report development errors.
EcuM_GoDown	Instructs the ECU State Manager module to perform a power off or a reset depending on the selected shutdown target.
EcuM_GoHalt	Instructs the ECU State Manager module to go into a sleep mode where the microcontroller is halted, depending on the selected shutdown target.
EcuM_GoPoll	Instructs the ECU State Manager module to go into a polling sleep mode depending on the selected shutdown target.
EcuM_SelectShutdownTarget	EcuM_SelectShutdownTarget selects the shutdown target. EcuM_SelectShutdownTarget is part of the ECU Manager Module port interface.
FrSM_AllSlots	This API function can be used to leave the KeySlotOnlyMode.
FrSM_SetEcuPassive	This API function can be used to set all FlexRay clusters of the ECU to a receive only mode.
J1939Dcm_SetState	Changes the communication state of J1939Dcm to offline or online.
J1939Rm_SetState	Changes the communication state of J1939Rm to offline (only Request for AC supported) or online.
LinSM_ScheduleRequest	The upper layer requests a schedule table to be changed on one LIN network.
Nm_DisableCommunication	Disables the NM PDU transmission ability. For that purpose <BusNm>_DisableCommunication shall be called (e.g. CanNm_DisableCommunication function is called if channel is configured as CAN).
Nm_EnableCommunication	Enables the NM PDU transmission ability. For that purpose <BusNm>_EnableCommunication shall be called (e.g. CanNm_EnableCommunication function is called if

	channel is configured as CAN).

⌋()

8.2 Service Interfaces

8.2.1 Sender-Receiver-Interfaces

8.2.1.1 AppModeRequestInterface

[SWS_BswM_000195] ⌈

Name	AppModeRequestInterface	
Comment	--	
IsService	true	
Variation	--	
Data Elements	requestedMode	
	Type	AppModeRequestType
	Variation	--

⌋()

8.2.2 Mode-Switch-Interfaces

8.2.2.1 AppModeInterface

[SWS_BswM_00196] ⌈

Name	AppModeInterface	
Comment	--	
IsService	true	
Variation	--	
ModeGroup	currentMode	AppMode

⌋()

8.2.3 Implementation Data Types

8.2.3.1 BswM_ModeType

[SWS_BswM_00197] ⌈

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

」()

8.2.3.2 BswM_UserType

[SWS_BswM_00198] 「

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

」()

8.2.3.3 BswM_ModeGroupType

[SWS_BswM_00199] 「

Name	BswM_ModeGroupType	
Kind	Type	
Derivedfrom	*BaseType*	*Variation*
	uint16	The range of valid IDs depends on configuration and on the chosen platform type.

	uint8	The range of valid IDs depends on configuration and on the chosen platform type.
Description	This type identifies the mode group that can be managed by a BswM.	
Range	0-255, 0-65535	The range of valid IDs depends on configuration and on the chosen platform type.
Variation	--	

」()

8.2.4 Ports

8.2.4.1 BswM_modeNotificationPort

[SWS_BswM_00200] 「

Name	modeNotificationPort_{Name}		
Kind	RequiredPort	Interface	AppModeInterface
Description	--		
Variation	Name = {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort/BswMModeRequestSource/BswMBswModeNotification.SHORT-NAME)}		

」()

8.2.4.2 BswM_modeRequestPort

[SWS_BswM_00201] 「

Name	modeRequestPort_{Name}		
Kind	RequiredPort	Interface	AppModeRequestInterface
Description	--		
Variation	Name = {ecuc(BswM/BswMConfig/BswMArbitration/BswMModeRequestPort.SHORT-NAME)}		

」()

8.2.4.3 BswM_modeSwitchPort

[SWS_BswM_00202] 「

Name	modeSwitchPort_{Name}		
Kind	ProvidedPort	Interface	AppModeInterface
Description	--		
Variation	Name = {ecuc(BswM/BswMConfig/BswMModeControl/BswMSwitchPort.SHORT-NAME)}		

」()

8.3 Call-back notifications

There are no call-back notifications in the BswM.

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] 「

Service name:	BswM_MainFunction

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

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

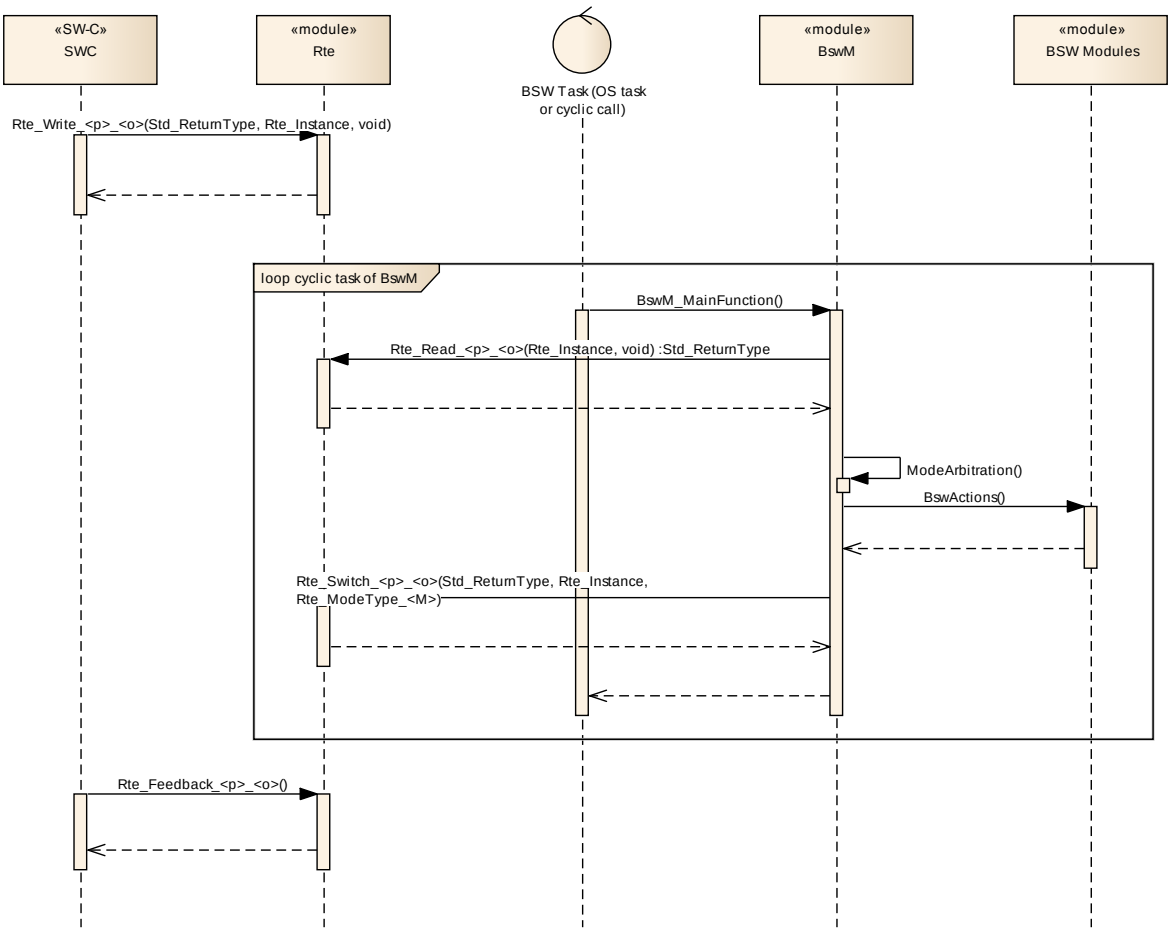
」()

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

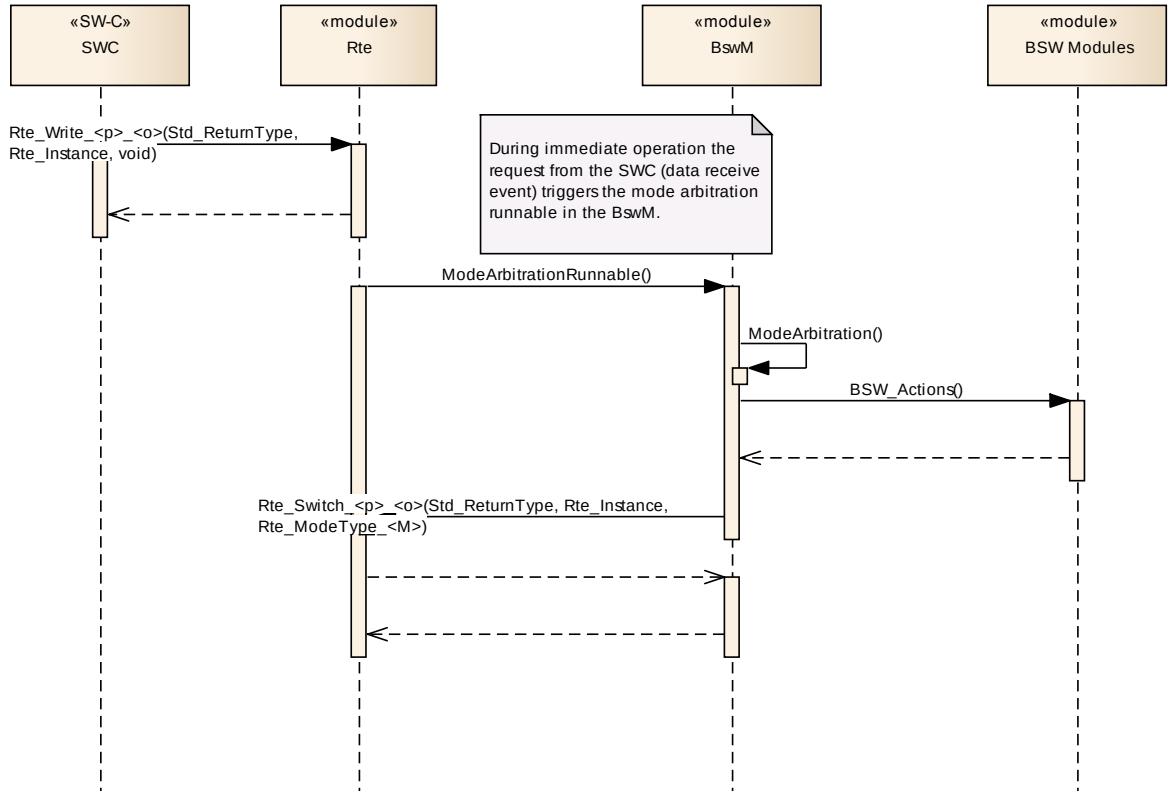
[SWS_BswM_00076] 「
If the BswMDevErrorDetect switch is enabled, the routine shall check if the BSW Mode Manager is initialized. If the BswM-mainfunction is uninitialized called from the BSW Scheduler, then it shall return immediately without performing any action and without reporting an error.」()

9 Sequence diagrams

9.1 Deferred operation of BswM



9.2 Immediate operation of BswM



10 Configuration specification

10.1 How to read this chapter

For details, refer to the related chapter in SWS_BSWGeneral.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. More detailed descriptions of the parameters are found in Chapters 7 and Chapter 8.

10.2.1 BswM

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

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMConfig	1..*	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition and post build configuration. This container does not have multipleConfigurationContainer=true due to compatibility issues in AUTOSAR R4.1. This restriction will no longer apply in R4.2. e.g. in an Ecu with two post build configurations and two partitions there would be 4 instances of the container BswMConfig, for all the combinations of partition and post build config one instance. Hint: the two instances (due to partitions) being part of the same post build config are referenced from the same instance of the EcuMConfiguration (via the reference EcuMFlexModuleConfigurationRef).
BswMGeneral	1	General configuration parameters of the Basic SW Mode Manager.

10.2.2 BswMConfig

SWS Item	**ECUC_BswM_00895 :**
Container Name	BswMConfig
Description	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module. This container exists once per partition and post build configuration. This container does not have multipleConfigurationContainer=true due to compatibility issues in AUTOSAR R4.1. This restriction will no longer apply in R4.2. e.g. in an Ecu with two post build configurations and two partitions there would be 4 instances of the container BswMConfig, for all the combinations of partition and post build config one instance. Hint: the two instances (due to partitions) being part of the same post build config are referenced from the same instance of the EcuMConfiguration (via the reference EcuMFlexModuleConfigurationRef).
Configuration Parameters	

SWS Item	**ECUC_BswM_00984 :**
Name	BswMPartitionRef
Description	This references the partition the BswM shall run inside.

Multiplicity	0..1		
Type	Reference to [EcucPartition]		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

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

10.2.3 BswMArbitration

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

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

		is evaluated to True or False.

10.2.4 BswMLogicalExpression

SWS Item	**ECUC_BswM_00808 :**
Container Name	BswMLogicalExpression
Description	This container describes the logical expressions that can be used for the mode arbitration. The logical expressions are built of a set of arguments and a logical operator. Each argument can either be a mode condition or a sub-expression to allow definition of more complex logical expressions. There may be an unlimited number of arguments in each logical expression. Note that the order of evaluation of the expressions is not defined. **Attributes:** postBuildChangeable=false
Configuration Parameters	

SWS Item	**ECUC_BswM_00814 :**		
Name	BswMLogicalOperator		
Description	This parameter specifies the logical operator to be used in the logical expression. If the expression only consists of a single condition this parameter shall not be used.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BSWM_AND	--	
	BSWM_NAND	--	
	BSWM_OR	--	
	BSWM_XOR	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00820 :**		
Name	BswMArgumentRef		
Description	This is a choice reference either to a mode condition or a sub-expression. In case the BswMLogicalExpression.BswMLogicalOperator equals BSWM_NAND only two operands are supported.		
Multiplicity	1..*		
Type	Choice reference to [BswMLogicalExpression , BswMModeCondition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.5 BswMModeCondition

SWS Item	**ECUC_BswM_00807 :**
Container Name	BswMModeCondition
Description	This container describes the BswM mode conditions that can be used either by itself to form a rule or as a part of a logical expression. **Attributes:** postBuildChangeable=false
Configuration Parameters	

SWS Item	**ECUC_BswM_00815 :**		
Name	BswMConditionType		
Description	This parameter specifies what kind of comparison that is made for the evaluation of the mode condition.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_EQUALS	--	
	BSWM_EQUALS_NOT	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00821 :**		
Name	BswMConditionMode		
Description	This parameter references the mode request port that is used for the condition.		
Multiplicity	1		
Type	Reference to [BswMModeRequestPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

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

10.2.6 BswMConditionValue

SWS Item	**ECUC_BswM_00816 :**
Choice container Name	BswMConditionValue
Description	This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***

BswMBswMode	0..1	This container defines the value and type of a mode in the BSW.
BswMModeDeclaration	0..1	When the mode corresponds to a mode request or mode indication interface the mode is defined by a mode declaration. The mode declarations are defined in the SW-C Template and hence a foreign reference to the corresponding Mode Declaration is used.

10.2.7 BswMBswMode

SWS Item	**ECUC_BswM_00869 :**
Container Name	BswMBswMode
Description	This container defines the value and type of a mode in the BSW.
Configuration Parameters	

SWS Item	**ECUC_BswM_00866 :**		
Name	BswMBswRequestedMode		
Description	This parameter contains the symbolic name (as a string) of a certain mode/state that can be requested/indicated by the BSW modules.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.8 BswMModeDeclaration

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

SWS Item	**ECUC_BswM_00864 :**
Name	BswMModeValueRef
Description	This is a foreign reference to the Mode Declaration used for the mode requests corresponding to this condition.
Multiplicity	1

Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.9 BswMModeRequestPort

SWS Item	**ECUC_BswM_00805 :**
Container Name	BswMModeRequestPort
Description	Each instance of this container defines a mode request interface that is used to requests or indicate modes from/to the BswM. These interfaces are implemented as ports or as ordinary C-functions based upon if the request is made by an SW-C or a BSW module. There are different types of mode requests: 1. Mode requests from the SW-C:s 2. Mode Requests from other BSW modules such as the DCM. 3. State/mode indications from the RTE or other BSW modules such as the bus specific State Managers. Note that the BswM treats all request and indications in the exact same way. **Attributes:** postBuildChangeable=false
Configuration Parameters	

SWS Item	**ECUC_BswM_00822 :**		
Name	BswMRequestProcessing		
Description	This parameter defines if the processing of the mode arbitration shall be done immediately when a mode request is received or if it shall be deferred to the processing of the main function of BswM.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_DEFERRED	--	
	BSWM_IMMEDIATE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMModeInitValue	0..1	This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. 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.

10.2.10 BswMModeInitValue

SWS Item	**ECUC_BswM_00928 :**
Container Name	BswMModeInitValue
Description	This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. This container is optional.
Configuration Parameters	

SWS Item	**ECUC_BswM_00932 :**		
Name	BswMBswModeInitValue		
Description	This parameter defines the initial mode value that is used by BswM for the corresponding mode request after initialization.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.11 BswMModeRequestSource

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

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMBswMModeRequest	0..1	The source of the mode request is a BswM on another partition. This container is only relevant for multi-partition systems.
BswMBswMModeSwitchNotification	0..1	This indicates that another BswM has switched a mode. This container is only relevant for multi-partition systems.
BswMBswModeNotification	0..1	This is a mode request source emanating from another BSW Module.
BswMCanSMIcomIndication	0..1	This is an indication from CanSM of the configuration Id of the Icom configuration
BswMCanSMIndication	0..1	This is an indication of the current state of the CAN State Manager.
BswMComMIndication	0..1	This is an indication of the current communication mode of a channel in the Communication Manager.

BswMComMInitiateReset	0..1	This is an indication from the ComM to signal a shutdown.
BswMComMPncRequest	0..1	This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager.
BswMDcmApplicationUpdatedIndication	0..1	This is a request to update application data from the DCM. This container does not contain any parameters since there are no further configuration needed for this type of request.
BswMDcmComModeRequest	0..1	The source of the mode request is the Diagnostic Communication Manager.
BswMEcuMIndication	0..1	This is a notification of the current operation mode of the ECU State Manager. This container does not contain any parameters since there are no further configuration needed for this type of request.
BswMEcuMWakeupSource	0..1	This is a notification of the current state of an ECU State Manager wakeup source.
BswMEthSMIndication	0..1	This is an indication of the current state of the Ethernet State Manager.
BswMFrSMIndication	0..1	This is an indication of the current state of the FlexRay State Manager.
BswMGenericRequest	0..1	This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers).
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.
BswMModeSwitchErrorEvent	0..1	This is a notification that an error occurred because the partition containing the mode users of the Mode Declaration Group Prototype was restarted by the RTE. Because the Mode Machine Instance holding the current mode can reside on that terminated partition, the Mode Manager has to be informed about the loss of this partition.
BswMNvMJobModeIndication	0..1	Indicates the current status of the multiblock job. The job is identified via BswMNvmService, e.g. 0x0c for NvmReadAll, 0x0d for NvmWriteAll. Possible Values are: NvM_RequestResultType NVM_REQ_OK NVM_REQ_NOT_OK NVM_REQ_PENDING NVM_REQ_INTEGRITY_FAILED NVM_REQ_BLOCK_SKIPPED NVM_REQ_NV_INVALIDATED NVM_REQ_CANCELED NVM_REQ_REDUNDANCY_FAILED NVM_REQ_RESTORED_FROM_ROM
BswMNvMRequest	0..1	Via this Mode Request Source the NvM indicates the current status of the specified

		block. Possible Values are: NvM_RequestResultType NVM_REQ_OK NVM_REQ_NOT_OK NVM_REQ_PENDING NVM_REQ_INTEGRITY_FAILED NVM_REQ_BLOCK_SKIPPED NVM_REQ_NV_INVALIDATED NVM_REQ_CANCELED NVM_REQ_REDUNDANCY_FAILED NVM_REQ_RESTORED_FROM_ROM
BswMPartitionRestarted	0..1	This is a notification that an error occured because the partition containing the BswM was restarted by the RTE. The Mode Users may lie in another (still running) partition. So the BswM has to be informed that the start of its partition is no normal startup but a restart of a single partition. This information can be used inside the Rules. This ModeIndication has to be used by the Restart Task of the particular partition.
BswMSdClientServiceCurrentState	0..1	Used by Service Discovery module to indicate current state of the Client Service (available/down).
BswMSdConsumedEventGroupCurrentState	0..1	Used by Service Discovery to indicate current status of the EventHandler (requested/released).
BswMSdEventHandlerCurrentState	0..1	Used by Service Discovery to indicate current status of the EventHandler (requested/released).
BswMSwcModeNotification	0..1	This is a mode switch notification associated with a RTE switch interface.
BswMSwcModeRequest	0..1	The source of the mode request is a SW Component.
BswMWdgMRequestPartitionReset	0..1	This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.

10.2.12 BswMBswMModeRequest

SWS Item	**ECUC_BswM_00980 :**
Container Name	BswMBswMModeRequest
Description	The source of the mode request is a BswM on another partition. This container is only relevant for multi-partition systems.
Configuration Parameters	

SWS Item	**ECUC_BswM_00981 :**		
Name	BswMBswMModeDeclarationGroupRef		
Description	This is a foreign reference to the Mode Declaration Group Prototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	

Scope / Dependency	scope: local

No Included Containers

10.2.13 BswMBswMModeSwitchNotification

SWS Item	**ECUC_BswM_00982 :**
Container Name	BswMBswMModeSwitchNotification
Description	This indicates that another BswM has switched a mode. This container is only relevant for multi-partition systems.
Configuration Parameters	

SWS Item	**ECUC_BswM_00983 :**		
Name	BswMBswMModeDeclarationGroupRef		
Description	This is a foreign reference to the Mode Declaration Group Prototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.14 BswMBswModeNotification

SWS Item	**ECUC_BswM_00926 :**
Container Name	BswMBswModeNotification
Description	This is a mode request source emanating from another BSW Module.
Configuration Parameters	

SWS Item	**ECUC_BswM_00927 :**		
Name	BswMBswModeDeclarationGroupPrototypeRef		
Description	This is a foreign reference to the Mode Declaration Group Prototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.15 BswMCanSMIcomIndication

SWS Item	**ECUC_BswM_01018 :**		
Container Name	BswMCanSMIcomIndication		
Description	This is an indication from CanSM of the configuration Id of the Icom configuration		
Configuration Parameters			

SWS Item	**ECUC_BswM_01020 :**		
Name	BswMCanSMIcomIndicationSwitchError		
Description	When this parameter is set to true, then this mode request source corresponds to error indications from the CanSM. Otherwise, this mode request source corresponds to error-free indications from the CanSM.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01019 :**		
Name	BswMCanSMChannelRef		
Description	This is a reference to the CAN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.16 BswMCanSMIndication

SWS Item	**ECUC_BswM_00857 :**
Container Name	BswMCanSMIndication
Description	This is an indication of the current state of the CAN State Manager.
Configuration Parameters	

SWS Item	**ECUC_BswM_00870 :**		
Name	BswMCanSMChannelRef		
Description	This is a reference to the CAN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD

	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.17 BswMComMIndication

SWS Item	**ECUC_BswM_00880 :**		
Container Name	BswMComMIndication		
Description	This is an indication of the current communication mode of a channel in the Communication Manager.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00883 :**		
Name	BswMComMChannelRef		
Description	This is a reference to the Communication Manager channel handle that the indication corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.18 BswMComMInitiateReset

SWS Item	**ECUC_BswM_01014 :**		
Container Name	BswMComMInitiateReset		
Description	This is an indication from the ComM to signal a shutdown.		
Configuration Parameters			

No Included Containers

10.2.19 BswMComMPncRequest

SWS Item	**ECUC_BswM_00922 :**		
Container Name	BswMComMPncRequest		
Description	This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00924 :**		
Name	BswMComMPncRef		
Description	This is a reference to the Communication Manager PNC handle of the Partial Network Cluster that the request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMPnc]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.20 BswMDcmApplicationUpdatedIndication

SWS Item	**ECUC_BswM_00925 :**
Container Name	BswMDcmApplicationUpdatedIndication
Description	This is a request to update application data from the DCM. This container does not contain any parameters since there are no further configuration needed for this type of request.
Configuration Parameters	

No Included Containers

10.2.21 BswMDcmComModeRequest

SWS Item	**ECUC_BswM_00863 :**
Container Name	BswMDcmComModeRequest
Description	The source of the mode request is the Diagnostic Communication Manager.
Configuration Parameters	

SWS Item	**ECUC_BswM_00876 : (Obsolete. Use BswMDcmComMChannelRef instead.)**
Name	BswMDcmComMNetwork
Description	This parameter is deprecated and will be removed in future. Please note that this parameter is replaced by BswMDcmComMChannelRef. Old description: This parameter specifies the network the request relates to. **Tags:** atp.Status=obsolete atp.StatusComment=This parameter is replaced by BswMDcmComMChannelRef. atp.StatusRevisionBegin=4.1.1
Multiplicity	0..1
Type	EcucStringParamDef

Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00969 :**		
Name	BswMDcmComMChannelRef		
Description	This is a reference from DcmModeRequest to the ComM channel that the indication corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.22 BswMEcuMIndication

SWS Item	**ECUC_BswM_00879 :**
Container Name	BswMEcuMIndication
Description	This is a notification of the current operation mode of the ECU State Manager. This container does not contain any parameters since there are no further configuration needed for this type of request.
Configuration Parameters	

No Included Containers

10.2.23 BswMEcuMWakeupSource

SWS Item	**ECUC_BswM_00904 :**
Container Name	BswMEcuMWakeupSource
Description	This is a notification of the current state of an ECU State Manager wakeup source.
Configuration Parameters	

SWS Item	**ECUC_BswM_00905 :**
Name	BswMEcuMWakeupSrcRef
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.24 BswMEthSMIndication

SWS Item	**ECUC_BswM_00860 :**
Container Name	BswMEthSMIndication
Description	This is an indication of the current state of the Ethernet State Manager.
Configuration Parameters	

SWS Item	**ECUC_BswM_00873 :**		
Name	BswMEthSMChannelRef		
Description	This is a reference to the Ethernet channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.25 BswMFrSMIndication

SWS Item	**ECUC_BswM_00858 :**
Container Name	BswMFrSMIndication
Description	This is an indication of the current state of the FlexRay State Manager.
Configuration Parameters	

SWS Item	**ECUC_BswM_00872 :**		
Name	BswMFrSMChannelRef		
Description	This is a reference to the FlexRay Cluster handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	

Scope / Dependency	scope: local

No Included Containers

10.2.26 BswMGenericRequest

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

SWS Item	**ECUC_BswM_00874 :**		
Name	BswMModeRequesterId		
Description	This parameters identifies the different users of the generic mode request interface. The allowable range of this parameter shall coincide with the range of BswM_UserType (which can be platform-dependent).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00875 :**		
Name	BswMRequestedModeMax		
Description	This parameter defines the upper limit for the modes requested by this mode requester. The allowable range of this parameter shall coincide with the range of BswM_ModeType (which can be platform-dependent).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.27 BswMJ1939DcmBroadcastStatus

SWS Item	**ECUC_BswM_00985 :**		
Container Name	BswMJ1939DcmBroadcastStatus		
Description	This is a notification of the desired broadcast status per network, triggered via DM13.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00988 :**		
Name	BswMJ1939DcmChannelRef		
Description	Reference to the channel represented by the nth bit in the networkMask parameter to J1939DcmBroadcastStatus.		
Multiplicity	1..*		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.28 BswMJ1939NmIndication

SWS Item	**ECUC_BswM_00966 :**		
Container Name	BswMJ1939NmIndication		
Description	This is an indication of the current state of the J1939 network management module.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00967 :**		
Name	BswMJ1939NmChannelRef		
Description	This is a reference to the J1939Nm channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00997 :**		
Name	BswMJ1939NmNodeRef		
Description	This is a reference to the node that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [J1939NmNode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.29 BswMLinSMIndication

SWS Item	**ECUC_BswM_00859 :**		
Container Name	BswMLinSMIndication		
Description	This is an indication of the current state of the LIN State Manager.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00871 :**		
Name	BswMLinSMChannelRef		
Description	This is a reference to the LIN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.30 BswMLinScheduleIndication

SWS Item	**ECUC_BswM_00885 :**		
Container Name	BswMLinScheduleIndication		
Description	This is an indication of the currently active LIN Schedule Table for a specific LIN Interface.		
Configuration Parameters			

SWS Item	**ECUC_BswM_01028 :**		
Name	BswMLinSMChannelRef		
Description	This is a reference to the LIN channel handle that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00886 :**
Name	BswMLinScheduleRef
Description	This is a reference to the LIN Schedule Table handle that the mode request corresponds to.

Multiplicity	1		
Type	Symbolic name reference to [LinSMSchedule]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.31 BswMLinTpModeRequest

SWS Item	**ECUC_BswM_00914 :**
Container Name	BswMLinTpModeRequest
Description	This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API.
Configuration Parameters	

SWS Item	**ECUC_BswM_00915 :**		
Name	BswMLinTpChannelRef		
Description	This is a reference to the LIN Interface Channel that the mode request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.32 BswMModeSwitchErrorEvent

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

SWS Item	**ECUC_BswM_00991 : (Obsolete)**
Name	BswMBswModeDeclarationGroupPrototypeRef
Description	Please note that this parameter is deprecated and will be removed in future. This is a foreign reference to the Mode Declaration Group Prototype where the ModeSwitchErrorEvent occurred.

	Tags: atp.Status=obsolete atp.StatusRevisionBegin=4.1.3		
Multiplicity	0..1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01030 :**		
Name	BswMRteSwitchPortRef		
Description	This is a reference to the BswMSwitchPort.		
Multiplicity	1		
Type	Reference to [BswMSwitchPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.33 BswMNvMJobModeIndication

SWS Item	**ECUC_BswM_00956 :**
Container Name	BswMNvMJobModeIndication
Description	Indicates the current status of the multiblock job. The job is identified via BswMNvmService, e.g. 0x0c for NvmReadAll, 0x0d for NvmWriteAll. Possible Values are: NvM_RequestResultType NVM_REQ_OK NVM_REQ_NOT_OK NVM_REQ_PENDING NVM_REQ_INTEGRITY_FAILED NVM_REQ_BLOCK_SKIPPED NVM_REQ_NV_INVALIDATED NVM_REQ_CANCELED NVM_REQ_REDUNDANCY_FAILED NVM_REQ_RESTORED_FROM_ROM
Configuration Parameters	

SWS Item	**ECUC_BswM_00957 :**		
Name	BswMNvmService		
Description	Identifies the Nvm job which is related to the mode request.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	NvmReadAll		NvmReadAll corresponds to service Id 0x0c
	NvmWriteAll		NvmWriteAll corresponds to service Id 0x0d
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.34 BswMNvMRequest

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

SWS Item	**ECUC_BswM_00891 :**		
Name	BswMNvMBlockRef		
Description	This is a reference to the NvM Block Descriptor that the request corresponds to.		
Multiplicity	1		
Type	Symbolic name reference to [NvMBlockDescriptor]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.35 BswMPartitionRestarted

SWS Item	**ECUC_BswM_00989 :**
Container Name	BswMPartitionRestarted
Description	This is a notification that an error occured because the partition containing the BswM was restarted by the RTE. The Mode Users may lie in another (still running) partition. So the BswM has to be informed that the start of its partition is no normal startup but a restart of a single partition. This information can be used inside the Rules. This ModeIndication has to be used by the Restart Task of the particular partition.
Configuration Parameters	

No Included Containers

10.2.36 BswMSdClientServiceCurrentState

SWS Item	**ECUC_BswM_01011 :**
Container Name	BswMSdClientServiceCurrentState
Description	Used by Service Discovery module to indicate current state of the Client

	Service (available/down).
Configuration Parameters	

SWS Item	**ECUC_BswM_001009 :**		
Name	BswMSdClientMethodsRef		
Description	This is a reference to a client service in the Sd module.		
Multiplicity	1		
Type	Symbolic name reference to [SdClientService]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.37 BswMSdConsumedEventGroupCurrentState

SWS Item	**ECUC_BswM_01012 :**
Container Name	BswMSdConsumedEventGroupCurrentState
Description	Used by Service Discovery to indicate current status of the EventHandler (requested/released).
Configuration Parameters	

SWS Item	**ECUC_BswM_001010 :**		
Name	BswMSdConsumedEventGroupRef		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.38 BswMSdEventHandlerCurrentState

SWS Item	**ECUC_BswM_01013 :**
Container Name	BswMSdEventHandlerCurrentState
Description	Used by Service Discovery to indicate current status of the EventHandler (requested/released).
Configuration Parameters	

SWS Item	**ECUC_BswM_001008 :**
Name	BswMSdEventHandlerRef
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]

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

No Included Containers

10.2.39 BswMSwcModeNotification

SWS Item	**ECUC_BswM_00892 :**
Container Name	BswMSwcModeNotification
Description	This is a mode switch notification associated with a RTE switch interface.
Configuration Parameters	

SWS Item	**ECUC_BswM_00893 :**		
Name	BswMSwcModeNotificationModeDeclarationGroupPrototypeRef		
Description	This is a foreign reference to the ModeDeclarationGroupPrototype.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.40 BswMSwcModeRequest

SWS Item	**ECUC_BswM_00862 :**
Container Name	BswMSwcModeRequest
Description	The source of the mode request is a SW Component.
Configuration Parameters	

SWS Item	**ECUC_BswM_00934 :**		
Name	BswMSwcModeRequestModeDeclarationGroupPrototypeRef		
Description	This is a foreign reference to the ModeDeclarationGroupPrototype used by the incoming mode request.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.41 BswMWdgMRequestPartitionReset

SWS Item	**ECUC_BswM_00916 :**
Container Name	BswMWdgMRequestPartitionReset
Description	This is a Partition Reset request from from the WdgM. This port corresponds to a call of the BswM_WdgM_RequestPartitionReset API.
Configuration Parameters	

SWS Item	**ECUC_BswM_00917 :**		
Name	BswMWdgMRequestPartitionResetRef		
Description	This is a reference to the partition that shall be reset.		
Multiplicity	1		
Type	Reference to [EcucPartition]		
ConfigurationClass	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.42 BswMRule

SWS Item	**ECUC_BswM_00806 :**
Container Name	BswMRule
Description	Each instance of this container describes a BswM arbitration rule. The rule either consists of a simple mode condition or a more complex logical expression. This container also references the action lists that shall be invoked when the rule is evaluated to True or False. **Attributes:** postBuildChangeable=true
Configuration Parameters	

SWS Item	**ECUC_BswM_00935 :**
Name	BswMNestedExecutionOnly
Description	This parameter defines for its related Rule if the Rule is an Independent rule or a Subordinate rule; false: an Independent rule, i.e. to be evaluated each time applicable (both as standalone Rule driven by its own BswMModeRequestSource and when referenced by another Rule). true: a Subordinated rule, to be evaluated ONLY as a result of being referenced in one or more Action Lists.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false

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

SWS Item	**ECUC_BswM_00888 :**		
Name	BswMRuleInitState		
Description	This parameter is a part of the reset/initialization behavior of BswM. Action lists are executed when the result of a rule evaluation have changed since the last evaluation. This parameter defines the "previous evaluation result" of a rule to be used after initialization of the BswM. If this parameter is set to BSWM_UNDEFINED, the evaluation result is always treated as changed at the first evaluation of the rule after initialization. If this parameter is set to BSWM_TRUE, the evaluation result is treated as changed if the rule is evaluated to false. If this parameter is set to BSWM_FALSE, the evaluation result is treated as changed if the rule is evaluated to true.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_FALSE	--	
	BSWM_TRUE	--	
	BSWM_UNDEFINED	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00819 :**		
Name	BswMRuleExpressionRef		
Description	This is a reference to the logical expression that is evaluated for each rule.		
Multiplicity	1		
Type	Reference to [BswMLogicalExpression]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00818 :**		
Name	BswMRuleFalseActionList		
Description	This is a reference to the action list that shall be executed when the rule is evaluated to False		
Multiplicity	0..1		
Type	Reference to [BswMActionList]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00817 :**
Name	BswMRuleTrueActionList
Description	This is a reference to the action list that shall be executed when the rule is evaluated to True
Multiplicity	0..1

Type	Reference to [BswMActionList]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.43 BswMDataTypeMappingSets

SWS Item **ECUC_BswM_00936 :**
Container Name BswMDataTypeMappingSets
Description Collection of references to DataTypeMappingSet.
Configuration Parameters

SWS Item	**ECUC_BswM_00937 :**		
Name	BswMDataTypeMappingSetRef		
Description	Reference to DataTypeMappingSet.		
Multiplicity	1..*		
Type	Foreign reference to [DATA-TYPE-MAPPING-SET]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.44 BswMModeControl

SWS Item **ECUC_BswM_00802 :**
Container Name BswMModeControl
Description This container includes all configuration sub-containers and parameters related to the mode control functionality of the BswM.
Configuration Parameters

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMAction	0..*	Each container of this type defines an action. These actions can be part of one or several action lists.
BswMActionList	0..*	Each instance of this container defines an action list that is invoked based on the BswM Rules. An action list contains a list of numbered action items to be processed. An action list can also include other action lists.
BswMRteModeRequestPort	0..*	This container defines a mode request port which the BswM may utilize to send a mode request to a SW-C which is acting as a mode-manager. If this container is referenced by a BswMRteModeRequest, the BswM shall create a

		corresponding PPort in its service description.
BswMSwitchPort	0..*	This container includes a reference to mode switch interface which the BswM must instantiate for the creation of a PPortPrototype in its SWCD.

10.2.45 BswMAction

SWS Item	**ECUC_BswM_00810 :**
Container Name	BswMAction
Description	Each container of this type defines an action. These actions can be part of one or several action lists. **Attributes:** postBuildChangeable=false
Configuration Parameters	

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

10.2.46 BswMAvailableActions

SWS Item	**ECUC_BswM_00826 :**
Choice container Name	BswMAvailableActions
Description	Choice container including the available actions to be used in the action lists.

Container Choices		
Container Name	***Multiplicity***	***Scope / Dependency***
BswMComMAllowCom	0..1	This container includes all parameters for the action to allow or to block communication for a ComM Channel. ComM_CommunicationAllowed is called when this action is configured.
BswMComMModeLimitation	0..1	This container includes all parameters related to a limitation of communication mode for a ComM Channel. ComM_LimitChannelToNoComMode is called when this action is configured.
BswMComMModeSwitch	0..1	This container includes all parameters related to a switch of communication mode for a ComM User. ComM_RequestComMode is called when this action is configured.
BswMCoreHaltMode	0..1	This container includes all parameters related to a switch of the activation state of core Halt.
BswMDeadlineMonitoringControl	0..1	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM.

		COM_ReceptionDMControl is called when this action is configured.
BswMEcuMGoDown	0..1	This container defines the UserId which shall be forwarded to the GoDown request.
BswMEcuMGoHalt	0..1	This container defines the action to trigger the EcuM_GoHalt from BswM.
BswMEcuMGoPoll	0..1	This container defines the action to trigger the EcuM_GoPoll from BswM.
BswMEcuMSelectShutdownTarget	0..1	This container defines the shutdown target.
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_EnableCommunication or Nm_DisableCommunication is called when this action is configured.
BswMPduGroupSwitch	0..1	This container includes references to the PDU groups that shall be enabled and disabled. Com_IpduGroupControl is called when this action is configured.
BswMPduRouterControl	0..1	This container includes all parameters related to enabling and disabling of routing of Routing Path Groups in the PDU Router. PduR_EnableRouting or PduR_DisableRouting is called when this action is configured.
BswMRequestRemoteMode	0..1	This container defines all information that are needed when a BswM wants to request a mode at another BswM which runs inside another partition.
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.
BswMSdClientServiceModeRequest	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.
BswMSdConsumedEventGroupModeRequest	0..1	This container includes all parameters related to the selection of a consumed EventGroup of Sd. Sd_ConsumedEventGroupSetState is called when this action is configured.
BswMSdServerServiceModeRequest	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.
BswMSwitchIPduMode	0..1	This container includes all parameters related to the selection of the transmission mode an I-PDU to be sent by COM. Com_SwitchIpduTxMode is called when this action is configured.
BswMTriggerIPduSend	0..1	This container includes all parameters related to the triggering of an I-PDU to be sent by COM. Com_TriggerIPDUSend is called when this action is configured.
BswMTriggerSlaveRTEStop	0..1	This container includes all parameters needed to stop the RTE on a slave core. This choice shall only be chosen if multicore is used.
BswMTriggerStartUpPhase2	0..1	This container includes all parameters needed to start phase two on a slave core. This choice shall only be chosen if multicore is used.
BswMUserCallout	0..1	This container includes all details needed for a user defined function call.

10.2.47 BswMComMAllowCom

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

SWS Item	**ECUC_BswM_00918 :**
Name	BswMComAllowed
Description	The parameter BswMComMAllowChannelRef refers to a channel which will allow or block communication using the function ComM_CommunicationAllowed(). This parameter corresponds to the parameter "Allowed" of the function ComM_CommunicationAllowed().
Multiplicity	1
Type	EcucBooleanParamDef
Default value	--

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

SWS Item	**ECUC_BswM_00912 :**		
Name	BswMComMAllowChannelRef		
Description	This is a reference to the ComM Channel for which communication shall be allowed or blocked. This reference corresponds to the parameter "Channel" of the function ComM_CommunicationAllowed().		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.48 BswMComMModeLimitation

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

SWS Item	**ECUC_BswM_00910 :**		
Name	BswMComMLimitMode		
Description	The function ComM_LimitChannelToNoComMode() takes in this boolean parameter to limit the channel's com mode to no-com mode. This parameter corresponds to the parameter "Status" of the function ComM_LimitChannelToNoComMode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00911 :**
Name	BswMComMLimitChannelRef
Description	This is a reference to the ComM channel for which the communication mode should be limited. This reference corresponds to the parameter "Channel" of the function

	ComM_LimitChannelToNoComMode.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.49 BswMComMModeSwitch

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

SWS Item	**ECUC_BswM_00840 :**		
Name	BswMComMRequestedMode		
Description	This parameter specifies if the requested communication mode. This parameter corresponds to the parameter "ComMode" of the function ComM_RequestComMode.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_FULL_COM	--	
	BSWM_NO_COM	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00841 :**		
Name	BswMComMUserRef		
Description	This is a reference to the ComM User that is associated to the Communication channel for which the communication mode should be requested. This reference corresponds to the parameter "User" of the function ComM_RequestComMode.		
Multiplicity	1		
Type	Symbolic name reference to [ComMUser]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.50 BswMCoreHaltMode

SWS Item	**ECUC_BswM_00970 :**
Container Name	BswMCoreHaltMode
Description	This container includes all parameters related to a switch of the activation state of core Halt.
Configuration Parameters	

SWS Item	**ECUC_BswM_00972 :**		
Name	BswMCoreHaltActivationState		
Description	Different possibilities are offered depending on the OS implementation and the CPU HW. The HALT modes addressed by this parameter are defined as names (strings) in the OS implementation. Different implementation may implement different HALT modes and subsequently different names.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00971 :**		
Name	BswMTargetCoreRef		
Description	This is a reference to the core on which the Core Halt process must be influenced.		
Multiplicity	1		
Type	Symbolic name reference to [EcucCoreDefinition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

Implementations of this BswM Action are suggested to utilize the OS API: ControlIdle()

10.2.51 BswMDeadlineMonitoringControl

SWS Item	**ECUC_BswM_00830 :**
Container Name	BswMDeadlineMonitoringControl
Description	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM.

	COM_ReceptionDMControl is called when this action is configured.
Configuration Parameters	

SWS Item	**ECUC_BswM_00852 :**		
Name	BswMDisabledDMPduGroupRef		
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be disabled. Together with the BswMEnabledDMPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function COM_ReceptionDMControl.		
Multiplicity	0..*		
Type	Symbolic name reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00851 :**		
Name	BswMEnabledDMPduGroupRef		
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be enabled. Together with the BswMDisabledDMPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function COM_ReceptionDMControl.		
Multiplicity	0..*		
Type	Symbolic name reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.52 BswMEcuMGoDown

SWS Item	**ECUC_BswM_00963 :**
Container Name	BswMEcuMGoDown
Description	This container defines the UserId which shall be forwarded to the GoDown request.
Configuration Parameters	

SWS Item	**ECUC_BswM_00964 :**		
Name	BswMEcuMUserIdRef		
Description	This is a reference to a EcuM UserId.		
Multiplicity	1		
Type	Symbolic name reference to [EcuMFlexUserConfig]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.53 BswMEcuMGoHalt

SWS Item	**ECUC_BswM_00995 :**
Container Name	BswMEcuMGoHalt
Description	This container defines the action to trigger the EcuM_GoHalt from BswM.
Configuration Parameters	

No Included Containers

10.2.54 BswMEcuMGoPoll

SWS Item	**ECUC_BswM_00996 :**
Container Name	BswMEcuMGoPoll
Description	This container defines the action to trigger the EcuM_GoPoll from BswM.
Configuration Parameters	

No Included Containers

10.2.55 BswMEcuMSelectShutdownTarget

SWS Item	**ECUC_BswM_00961 :**
Container Name	BswMEcuMSelectShutdownTarget
Description	This container defines the shutdown target.
Configuration Parameters	

SWS Item	**ECUC_BswM_00993 :**		
Name	BswMEcuMShutdownTarget		
Description	This parameter contains the shutdown target that the BswM selects at the EcuM.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	OFF	--	
	RESET	In case the configuration parameter BswMEcuMShutdownTarget is set to RESET the configuration parameter BswMEcuMResetModeRef shall exist and contain a valid reference to a EcuM reset mode.	
	SLEEP	In case the configuration parameter BswMEcuMShutdownTarget is set to SLEEP the configuration parameter BswMEcuMSleepModeRef shall exist and contain a valid reference to a EcuM sleep mode.	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-

			BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00994 :**		
Name	BswMEcuMResetModeRef		
Description	This is a reference to a reset mode.		
Multiplicity	0..1		
Type	Symbolic name reference to [EcuMResetMode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00962 :**		
Name	BswMEcuMSleepModeRef		
Description	This is a reference to a sleep mode.		
Multiplicity	0..1		
Type	Symbolic name reference to [EcuMSleepMode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.56 BswMJ1939DcmStateSwitch

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

SWS Item	**ECUC_BswM_01035 :**		
Name	BswMJ1939DcmRequestedState		
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	J1939DCM_STATE_OFFLINE	--	
	J1939DCM_STATE_ONLINE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01033 :**		
Name	BswMJ1939DcmChannelRef		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01034 :**		
Name	BswMJ1939DcmNodeRef		
Description	This reference points to a J1939NmNode and provides access to the unique J1939NmNodeId.This reference corresponds to the parameter "node" of the function J1939Dcm_SetState.		
Multiplicity	1		
Type	Symbolic name reference to [J1939NmNode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.57 BswMJ1939RmStateSwitch

SWS Item	**ECUC_BswM_00998 :**
Container Name	BswMJ1939RmStateSwitch
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.
Configuration Parameters	

SWS Item	**ECUC_BswM_01002 :**		
Name	BswMJ1939RmRequestedState		
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	J1939RM_STATE_OFFLINE	--	
	J1939RM_STATE_ONLINE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01000 :**		
Name	BswMJ1939RmChannelRef		
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_SetState.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01001 :**		
Name	BswMJ1939RmNodeRef		
Description	This reference points to a J1939NmNode and provides access to the unique J1939NmNodeId. This reference corresponds to the parameter "node" of the function J1939Rm_SetState.		
Multiplicity	1		
Type	Symbolic name reference to [J1939NmNode]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.58 BswMLinScheduleSwitch

SWS Item	**ECUC_BswM_00827 :**		
Container Name	BswMLinScheduleSwitch		
Description	This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured. The configuration for the "network" parameter can be accessed via the reference LinSMComMNetworkHandleRef contained in the parent container LinSMChannel of the container referenced by BswMLinScheduleRef.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00842 :**		
Name	BswMLinScheduleRef		
Description	This is a reference to the LIN schedule table that the LIN SM shall change to. This reference corresponds to the parameter "schedule" of the function LinSM_ScheduleRequest.		
Multiplicity	1		
Type	Symbolic name reference to [LinSMSchedule]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME

	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.59 BswMNMControl

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

SWS Item	**ECUC_BswM_00838 :**		
Name	BswMNMAction		
Description	This parameter specifies if the communication of the corresponding NM channel should be enabled or disabled.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_NM_DISABLE	--	
	BSWM_NM_ENABLE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00999 :**		
Name	BswMComMNetworkHandleRef		
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId. This reference corresponds to the parameter "NetworkHandle" of the function Nm_EnableCommunication and Nm_DisableCommunication.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.60 BswMPduGroupSwitch

SWS Item	**ECUC_BswM_00828 :**		
Container Name	BswMPduGroupSwitch		
Description	This container includes references to the PDU groups that shall be enabled and disabled. Com_IpduGroupControl is called when this action is configured.		
Configuration Parameters			

SWS Item	**ECUC_BswM_00913 :**		
Name	BswMPduGroupSwitchReinit		
Description	This parameter defines if the values for the parameters like periodical timer, minimum delay timer etc is retainer or reinitialized during a PDU Group Switch. This parameter corresponds to the parameter "initialize" of the function Com_IpduGroupControl.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00850 :**		
Name	BswMDisabledPduGroupRef		
Description	This is a reference to a PDU Group that should be disabled. Together with the BswMEnabledIPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function Com_IpduGroupControl.		
Multiplicity	0..*		
Type	Symbolic name reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00849 :**		
Name	BswMEnabledPduGroupRef		
Description	This is a reference to a PDU Group that should be enabled. Together with the BswMDisabledIPduGroupRef this reference corresponds to the parameter "ipduGroupVector" of the function Com_IpduGroupControl.		
Multiplicity	0..*		
Type	Symbolic name reference to [ComIPduGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.61 BswMPduRouterControl

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

SWS Item	**ECUC_BswM_00854 :**		
Name	BswMPduRouterAction		
Description	This parameter specifies if the routing of the corresponding PDU should be enabled or disabled.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BSWM_PDUR_DISABLE	--	
	BSWM_PDUR_ENABLE	--	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00855 :**		
Name	BswMPduRoutingPathGroupRef		
Description	This is a reference to the PDU Routing Path Group for which the routing in the PDU Router should be enabled or disabled. This reference corresponds to the parameter "id" of the function PduR_EnableRouting and PduR_DisableRouting.		
Multiplicity	1..*		
Type	Symbolic name reference to [PduRRoutingPathGroup]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.62 BswMRequestRemoteMode

SWS Item	**ECUC_BswM_00976 :**
Container Name	BswMRequestRemoteMode
Description	This container defines all information that are needed when a BswM wants to request a mode at another BswM which runs inside another partition.
Configuration Parameters	

SWS Item	**ECUC_BswM_00977 :**		
Name	BswMEcuCPartitionRef		
Description	This is a reference to the EcuCPartition the other BswM belongs to.		
Multiplicity	1		
Type	Reference to [EcucPartition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-

			BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00979 :**		
Name	BswMModeDeclarationGroupPrototypeRef		
Description	This is the reference to the ModeDeclarationGroup Prototype for that a mode shall be requested at another BswM.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00978 :**		
Name	BswMRequestedRemoteMode		
Description	This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

Implementations of this BswM Action are suggested to utilize this BswM API: BswM_BswMModeRequest ()

10.2.63 BswMRteModeRequest

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

SWS Item	**ECUC_BswM_01024 :**		
Name	BswMRequestedModeRef		
Description	This is a foreign reference to the Mode Declaration used for the mode request		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD

	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01023 :**		
Name	BswMRteModeRequestPortRef		
Description	This is a reference to a BswMRteModeRequestPort.		
Multiplicity	1		
Type	Reference to [BswMRteModeRequestPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.64 BswMRteSwitch

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

SWS Item	**ECUC_BswM_00952 :**		
Name	BswMRteSwitchPortRef		
Description	This is a reference to the BswMSwitchPort.		
Multiplicity	1		
Type	Reference to [BswMSwitchPort]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00896 :**		
Name	BswMSwitchedMode		
Description	This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.65 BswMSchMSwitch

SWS Item	**ECUC_BswM_00899 :**		
Container Name	BswMSchMSwitch		
Description	This container defines a mode switch indication that the BswM provides to the SW-C that need to be notified about the mode switch. SchM_Switch is called when this action is configured.		
Configuration Parameters			

SWS Item	**ECUC_BswM_01031 :**		
Name	BswMSchMModeDeclarationGroupRef		
Description	This is the reference to a ModeDeclarationGroup to define a ModeDeclarationGroupPrototype in the role BswModuleDescription.providedModeGroup.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION-GROUP]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00901 :**		
Name	BswMSchMSwitchedMode		
Description	This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group.		
Multiplicity	1		
Type	Foreign reference to [MODE-DECLARATION]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

[SWS_BswM_00219]⌈ If a BswMSchMSwitch is configured, then the BswM shall create the BswModuleDescription.providedModeGroup aggregation in its BSWMD.⌋()

10.2.66 BswMSdClientServiceModeRequest

SWS Item	**ECUC_BswM_000974 :**
Container Name	BswMSdClientServiceModeRequest
Description	This container includes all parameters related to the selection of an client service of Sd. Sd_ClientServiceSetState is called when this action is configured.
Configuration Parameters	

SWS Item	**ECUC_BswM_01016 :**
Name	BswMSdClientServiceState
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	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_001009 :**		
Name	BswMSdClientMethodsRef		
Description	This is a reference to a client service in the Sd module.		
Multiplicity	1		
Type	Symbolic name reference to [SdClientService]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.67 BswMSdConsumedEventGroupModeRequest

SWS Item	**ECUC_BswM_001004 :**
Container Name	BswMSdConsumedEventGroupModeRequest
Description	This container includes all parameters related to the selection of a consumed EventGroup of Sd. Sd_ConsumedEventGroupSetState is called when this action is configured.
Configuration Parameters	

SWS Item	**ECUC_BswM_01017 :**		
Name	BswMSdConsumedEventGroupState		
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.	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_001010 :**		
Name	BswMSdConsumedEventGroupRef		
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]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.68 BswMSdServerServiceModeRequest

SWS Item	**ECUC_BswM_001005 :**
Container Name	BswMSdServerServiceModeRequest
Description	This container includes all parameters related to the selection of a server service of Sd. Sd_ServerServiceSetState is called when this action is configured.
Configuration Parameters	

SWS Item	**ECUC_BswM_01015 :**		
Name	BswMSdServerServiceState		
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.
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_001007 :**		
Name	BswMSdServerMethodsRef		
Description	This is a reference to a server service in the Sd module.		
Multiplicity	1		
Type	Symbolic name reference to [SdServerService]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.69 BswMSwitchIPduMode

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

SWS Item	**ECUC_BswM_00960 :**		
Name	BswMSwitchIPduModeValue		
Description	This parameter defines which transmission mode shall be selected during this call. This parameter corresponds to the parameter "truefalsemode" of the function Com_SwitchIpduTxMode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00959 :**		
Name	BswMSwitchIPduModeRef		
Description	This is a reference to an I-PDU for which the transmission mode shall be set. This reference corresponds to the parameter "PduId" of the function Com_SwitchIpduTxMode.		
Multiplicity	1		
Type	Symbolic name reference to [ComIPdu]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.70 BswMTriggerIPduSend

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

SWS Item	**ECUC_BswM_00907 :**
Name	BswMTriggeredIPduRef

Description	This is a reference to an I-PDU that should be triggered for transmission. This reference corresponds to the parameter "PduId" of the function Com_TriggerIPDUSend.		
Multiplicity	1..*		
Type	Symbolic name reference to [ComIPdu]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.71 BswMTriggerSlaveRTEStop

SWS Item	**ECUC_BswM_00919 :**
Container Name	BswMTriggerSlaveRTEStop
Description	This container includes all parameters needed to stop the RTE on a slave core. This choice shall only be chosen if multicore is used.
Configuration Parameters	

SWS Item	**ECUC_BswM_00921 : (Obsolete)**		
Name	BswMCoreId		
Description	This parameter is deprecated and will be removed in future. Please note that this parameter is replaced by BswMCoreRef. Old description: This parameter defines the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter. **Tags:** atp.Status=obsolete atp.StatusComment=This parameter is replaced by BswMCoreRef atp.StatusRevisionBegin=4.1.1		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00973 :**		
Name	BswMCoreRef		
Description	This reference points to the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter.		
Multiplicity	0..1		
Type	Symbolic name reference to [EcucCoreDefinition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-

			BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.72 BswMTriggerStartUpPhase2

SWS Item	**ECUC_BswM_00920 :**
Container Name	BswMTriggerStartUpPhase2
Description	This container includes all parameters needed to start phase two on a slave core. This choice shall only be chosen if multicore is used.
Configuration Parameters	

SWS Item	**ECUC_BswM_00921 : (Obsolete)**		
Name	BswMCoreId		
Description	This parameter is deprecated and will be removed in future. Please note that this parameter is replaced by BswMCoreRef. Old description: This parameter defines the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter. **Tags:** atp.Status=obsolete atp.StatusComment=This parameter is replaced by BswMCoreRef atp.StatusRevisionBegin=4.1.1		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00973 :**		
Name	BswMCoreRef		
Description	This reference points to the identifier of the slave core that is used as input parameter for the BswM_TriggerStartUpPhase2 and BswM_TriggerSlaveRTEStop functions. The value of this parameter shall be synchronized with the OsApplicationCoreAssignment parameter.		
Multiplicity	0..1		
Type	Symbolic name reference to [EcucCoreDefinition]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.73 BswMUserCallout

SWS Item	**ECUC_BswM_00834 :**
Container Name	BswMUserCallout
Description	This container includes all details needed for a user defined function call.
Configuration Parameters	

SWS Item	**ECUC_BswM_00843 :**		
Name	BswMUserCalloutFunction		
Description	This parameter specifies the complete function call including all parameters. The parameters are specified during configuration time, and cannot be changed during run time. Any return values passed by the callout will be ignored. Example usage can be: Actions to initialize other BSW modules Action to call Rte_Start() Action to call Rte_Stop() Action to call NvM_ReadAll() Action to call NvM_WriteAll()		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.74 BswMActionList

SWS Item	**ECUC_BswM_00809 :**
Container Name	BswMActionList
Description	Each instance of this container defines an action list that is invoked based on the BswM Rules. An action list contains a list of numbered action items to be processed. An action list can also include other action lists. **Attributes:** postBuildChangeable=true
Configuration Parameters	

SWS Item	**ECUC_BswM_00894 :**
Name	BswMActionListExecution
Description	This parameter controls if the corresponding action list shall be executed every time the rule is evaluated or only when the result of the evaluation changes. 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.	
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

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

10.2.75 BswMActionListItem

SWS Item	**ECUC_BswM_00823 :**
Container Name	BswMActionListItem
Description	This container defines an item in an action list. **Attributes:** postBuildChangeable=true
Configuration Parameters	

SWS Item	**ECUC_BswM_00902 :**		
Name	BswMAbortOnFail		
Description	This parameter defines if the execution of the action list shall be aborted if this specific action returns E_NOT_OK. Note that this is only applicable for actions that have E_NOT_OK as a possible return value.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00824 :**		
Name	BswMActionListItemIndex		
Description	This parameter defines the index of the action in the action list. It is used define in which order the actions shall be performed.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

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

SWS Item	**ECUC_BswM_00933 :**		
Name	BswMReportFailToDemRef		
Description	If the reference is given, the DEM event shall be reported failed if this specific action returns E_NOT_OK; it shall be reported passed if this specific action returns E_OK.		
Multiplicity	0..1		
Type	Symbolic name reference to [DemEventParameter]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.76 BswMRteModeRequestPort

SWS Item	**ECUC_BswM_01022 :**		
Container Name	BswMRteModeRequestPort		
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. **Attributes:** postBuildChangeable=false		
Configuration Parameters			

SWS Item	**ECUC_BswM_01027 :**		
Name	BswMRteModeRequestPortInterfaceMappingRef		
Description	This is a foreign reference to the variable and parameter interface mapping used for the mode request.		
Multiplicity	0..1		
Type	Foreign reference to [VARIABLE-AND-PARAMETER-INTERFACE-MAPPING]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01025 :**		
Name	BswMRteModeRequestPortInterfaceRef		
Description	This is an instance reference to the variable data prototype used for the mode request.		
Multiplicity	1		
Type	Instance reference to [VARIABLE-DATA-PROTOTYPE context: SW-COMPONENT-PROTOTYPE* PORT-PROTOTYPE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.77 BswMSwitchPort

SWS Item	**ECUC_BswM_00950 :**
Container Name	BswMSwitchPort
Description	This container includes a reference to mode switch interface which the BswM must instantiate for the creation of a PPortPrototype in its SWCD. **Attributes:** postBuildChangeable=false
Configuration Parameters	

SWS Item	**ECUC_BswM_00951 :**		
Name	BswMModeSwitchInterfaceRef		
Description	Reference to the ModeSwitchInterface of this BswMModeSwitchPort.		
Multiplicity	1		
Type	Foreign reference to [MODE-SWITCH-INTERFACE]		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.78 BswMGeneral

SWS Item	**ECUC_BswM_00800 :**
Container Name	BswMGeneral
Description	General configuration parameters of the Basic SW Mode Manager.
Configuration Parameters	

SWS Item	**ECUC_BswM_00938 :**
Name	BswMCanSMEnabled {BSWM_CANSM_ENABLED}

Description	enable/disable CanSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_01029 :**		
Name	BswMCanSMIcomEnabled {BSWM_CANSM_ENABLED}		
Description	enable/disable CanSM Icom related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00939 :**		
Name	BswMComMEnabled {BSWM_COMM_ENABLED}		
Description	enable/disable ComM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00940 :**		
Name	BswMDcmEnabled {BSWM_DCM_ENABLED}		
Description	enable/disable Dcm module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00811 :**		
Name	BswMDevErrorDetect {BSWM_DEV_ERROR_DETECT}		
Description	Switches the Development Error Detection and Notification ON or OFF. true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	

Scope / Dependency	scope: local

SWS Item	**ECUC_BswM_00941 :**		
Name	BswMEcuMEnabled {BSWM_ECUM_ENABLED}		
Description	enable/disable EcuM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00942 :**		
Name	BswMEthSMEnabled {BSWM_ETHSM_ENABLED}		
Description	enable/disable EthSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00943 :**		
Name	BswMFrSMEnabled {BSWM_FRSM_ENABLED}		
Description	enable/disable FrSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00949 :**		
Name	BswMGenericRequestEnabled {BSWM_GENERIC_REQUEST_ENABLED}		
Description	enable/disable Generic Request related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00987 :**
Name	BswMJ1939DcmEnabled {BSWM_J1939DCM_ENABLED}
Description	Enable/disable J1939Dcm module related BswM API: true: Enabled false: Disabled
Multiplicity	0..1

Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00965 :**		
Name	BswMJ1939NmEnabled {BSWM_J1939NM_ENABLED}		
Description	Enable/disable J1939Nm module related BswM API. true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00944 :**		
Name	BswMLinSMEnabled {BSWM_LINSM_ENABLED}		
Description	enable/disable LinSM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00945 :**		
Name	BswMLinTPEnabled {BSWM_LINTP_ENABLED}		
Description	enable/disable LinTP module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_BswM_00813 :**		
Name	BswMMainFunctionPeriod		
Description	The cycle time of the periodic main function of BswM. Defined in seconds .		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. INF		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_BswM_00946 :		
Name	BswMNvMEnabled {BSWM_NVM_ENABLED}		
Description	enable/disable NvM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_BswM_00947 :		
Name	BswMSchMEnabled {BSWM_SCHM_ENABLED}		
Description	enable/disable SchM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_BswM_00968 :		
Name	BswMSdEnabled {BSWM_SD_ENABLED}		
Description	enable/disable Sd module related BswM_Sd_CurrentState API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_BswM_00812 :		
Name	BswMVersionInfoApi		
Description	Switches the possibility to read the version information with the service BswM_GetVersionInfo(). true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
ConfigurationClass	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	ECUC_BswM_00948 :		
Name	BswMWdgMEnabled {BSWM_WDGM_ENABLED}		
Description	enable/disable WdgM module related BswM API: true: Enabled false: Disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		

ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

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

10.2.79 BswMUserIncludeFiles

SWS Item	**ECUC_BswM_00954 :**
Container Name	BswMUserIncludeFiles
Description	Collection of header file names which shall be included by the BswM.
Configuration Parameters	

SWS Item	**ECUC_BswM_00955 :**		
Name	BswMUserIncludeFile		
Description	Header file name which shall be included by the BswM. The value of this parameter shall be used as h-char-sequence or q-char-sequence according to ISO C90 section 6.10.2 "source file inclusion". The parameter value MUST NOT represent a path, since ISO C90 does not specify how such a path is treated (i.e., this is implementation defined (and additionally depends on the operating system and the underlying file system)).		
Multiplicity	1..*		
Type	EcucStringParamDef		
Default value	--		
maxLength	--		
minLength	--		
regularExpression	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.3 Published Information

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

11 Not applicable requirements

[SWS_BswM_09999]「 These requirements are not applicable to this specification. 」
(SRS_BSW_00405, SRS_BSW_00170, SRS_BSW_00387, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00336, SRS_BSW_00339, SRS_BSW_00409)